REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to the Department of Defense, Executive Services and Communications Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 15-09-2008	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* 4 May 2006 – 31 August 2008

4. TITLE AND SUBTITLE Hierarchical High Level Information Fusion (H2LIFT)	**5a. CONTRACT NUMBER** N00014-06-C-0019
	5b. GRANT NUMBER N/A
	5c. PROGRAM ELEMENT NUMBER N/A
6. AUTHOR(S) Sudit, Moises Stotz, Adam Crassidis, John Crassidis, Aggamemnon Nagi, Rakesh VonGonten, Tracy	**5d. PROJECT NUMBER** N/A
	5e. TASK NUMBER N/A
	5f. WORK UNIT NUMBER N/A
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **CUBRC, Inc.** 4455 Genesee Street, Buffalo, NY 14225 **University at Buffalo** 402 Crofts Hall, Amherst, NY 14260 **L3 Communications-IS** 10001 Jack Finney Blvd, Greenville, TX 75402 **Rochester Institute of Technology** 141 Lomb Memorial Drive, Suite 2000, Rochester, NY 14623	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research 875 North Randolph St., Suite 1425 Arlington, VA 22203-1995	**10. SPONSOR/MONITOR'S ACRONYM(S)** ONR
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Unlimited/Unclassified

13. SUPPLEMENTARY NOTES

N/A

14. ABSTRACT

The primary objective of this effort was the progression of Level 2/3 fusion of informational content to obtain an advanced multi-intelligent system for hierarchical high-level decision making processes. The goal was to develop an information integration mechanism to simplify human decision making solving operational problems. As technology continues to advance and the proliferation of sensors in all platform increases, human decision makers are being overwhelmed with data. In this research, the CUBRC proposed a cost effective two-year program of a novel approach in the near "real-time" ranking/formulation of hypotheses in asymmetric warfare scenarios. In particular, CUBRC introduced the Hierarchical High Level Information Fusion Technologies (H^2LIFT) architecture with the following objectives:

- develop H^2LIFT Architecture and algorithms for GWOT/MDA threats,
- develop prototype software that implements H^2LIFT architecture and algorithms, and
- develop a simulation based tool for performance evaluation and analysis

15. SUBJECT TERMS

Tracking, Situation Assessment, Level1, 2 Distributed Information Fusion, Contextual Data Simulation, Anomaly Visualization

16. SECURITY CLASSIFICATION OF: U			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Moises Sudit
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	42	**19b. TELEPHONE NUMBER** *(include area code)* (716) 204-5155

CUBRC®

Via email: wendy.martinez@navy.mil

16 September 2008
AML08-062

Dr. Wendy Martinez, Code 311
Office of Naval Research
875 N Randolph St, Suite 1425
Arlington, VA 22203-1995

Subject: Contract No. N00014-06-C-0019
Hierarchical High Level Information Fusion (H^2LIFT)

Enclosure: Final Report (1 hard copy and 1 DVD)

Dear Dr. Martinez:

In accordance with the requirements of Contract No. N00014-06-C-0019, CUBRC, Inc. herein submits the enclosed Final Report and a DVD containing source code, executable code and user's manual.

This package will also arrive in the mail.

Should additional technical information be needed, please contact Mr. Michael Moskal by telephone at 716-204-5120 or by e-mail at Moskal@cubrc.org or Dr. Moises Sudit by telephone at 716-204-5155 or by email at Sudit@cubrc.org. Questions of an administrative or contractual nature should be directed to Mr. Kevin W. Leous by telephone at 716-204-5137 or by email at Leous@cubrc.org or to me by telephone at 716-204-5124 or by email at Lyte@cubrc.org.

Regards,
CUBRC, Inc.
Andel Lyte
Mr. Andel Lyte
Contract Administrator

Cc:
Nicholas Anderson (via nicholas.anderson1@navy.mil)

Final Report for the Hierarchical High Level Information Fusion Program (H^2LIFT)

Prime Contract No. N00014-06-C-0019

CUBRC Project No. 06823

Data Item A005

Final Report

Submitted to:

Wendy Martinez

Office of Naval Research

875 North Randolph St., Suite 1425
Arlington, VA 22203-1995

Submitted by:

CUBRC, Inc.

4455 Genesee Street
Buffalo, NY 14225

15 September 2008

Table of Contents

1 INTRODUCTION

This document is written to satisfy Data Item No. A005, the Final Report for the Hierarchical High Level Information Fusion Program (H^2Lift). This contract, Prime Contract Number N00014-06-C-0019, had a start date of 4 May 2006 and an original end date of 3 May 2008. However, Contract Amendment / Modification No. P00004 extended the period of performance through 31 August 2008.

2 REFERENCES

1. Office of Naval Research Award/Contract No. N00014-06-C-0019
2. J. Crassidis, *Sigma-Point Kalman Filtering for Integrated GPS and Inertial Navigation*, American Institute of Aeronautics and Astronautics.
3. P. Dana, http://www.ncgia.ucsb.edu/giscc/units/u015/u015.htm
4. Gainer, Hoffman, Summary of Transformation Equations of Motion Used in Free-Flight and Wind Tunnel Data Reduction and Analysis, Langly Research Center, 1972.
5. E. Williams, http://williams.best.vwh.net/avform.htm
6. E. Williams, http://williams.best.vwh.net/ellipsoid/node3.html
7. Bar-Shalom, Y. and Fortmann, T.E., Tracking and Data Association, Academic Press, Boston, MA, 1988.
8. Crassidis, J.L., and Junkins, J.L., Optimal Estimation of Dynamical Systems, Chapman & Hall/CRC, Boca Raton, FL, 2004.
9. Crassidis, J.L., and Cheng, Y., "Generalized Multiple-Model Adaptive Estimation Using an Autocorrelation Approach," 9th International Conference on Information Fusion, Florence, Italy, July 2006, paper 223.
10. "The National Strategy for Maritime Security", 2005.
11. J. Morgan and B. Wimmer, "Enhancing Awareness in the Maritime Domain", CHIPS Magazine, 14-15, 2005.
12. "The National Plan to Achieve Maritime Domain Awareness", 2005.
13. M. Balci and R. Pegg, "Toward Global Maritime Awareness – A Technical Perspective", Proceedings of the 9th International Conference on Information Fusion, 2006.
14. W. Kay, "Global maritime intelligence integration / maritime domain awareness", Presented at the Maritime Security Conference, 2007
15. L. Belfares and A. Guitouni, "Multi-Objective Genetic Algorithms for Course of Action Planning", Proceedings of the 2003 Congress on Evolutionary Computation, Volume 3, pp. 1543-1551, 2003.
16. J. Hill, et al., "Tactical Event Resolution using Software Agents, Crisp Rules, and a Genetic Algorithm", Proceedings of the Advanced Simulation Technologies Conference, pp. 15 – 21, 2000.
17. R. Mittu and J. Walters, "Course of Action Analysis in the Global Information Grid", The NRL Review, pp. 160-162, 2005.

18. W. Cairns, “On Watch: Vessel Tracking Technologies for Maritime Security”, USCG Proceedings, Spring, 2006.

19. World Shipping Register web site, http://www.world-register.org/.

20. U.S. Department of Defense, Data Fusion SubPanel of the Joint Directors of Laboratories, “Technical panel for c3”, Data Fusion Lexicon, October, 1991.

21. D. L. Hall and J. Llinas, “An introduction to multisensor data fusion”, Proceedings of the IEEE, vol. 85, no. 1, January 1997, pp. 6–23.

22. J. Llinas, et al., “Revisions and extensions to the JDL data fusion model II”, Proceedings of The 7th International Conference on Information Fusion, June 2004, pp. 1218–1230.

23. A. Stotz and M. Sudit, “INformation Fusion Engine for Real-time Decision Making (INFERD): A Perceptual System for Cyber Attack Tracking”, Proceedings of the 10th International conference on Information Fusion, July 2007.

24. G. Pulford, et al., “Taxonomy of multiple target tracking methods”, IEEE Proceedings of Radar, Sonar, and Navigation, vol. 152, 2005.

25. Mark Musen, “Dimensions of knowledge sharing and reuse”, Computers and biomedical research, vol. 25, no. 5, 1992, pp. 435-467.

26. R. Patil, et al., “The DARPA knowledge sharing effort: progress report”, Readings in Agents, Morgan Kaufmann Publishers Inc., 1997, pp. 243-254.

27. H. Shiow-Fen and J. C. Gerard, “k-neighborhood-covering and –independence problems for chordal graphs”, SIAM Journal on Discrete Mathematics, vol. 11, no. 4, pp. 633-643, 1998.

28. S. Fujita, “Neighborhood information dissemination in the star graph”, IEEE Transactions on Computers, vol. 49, no. 12, pp. 1366–1370, 2000.

29. A. Schenker, et al., “Classification of web documents using graph matching”, International Journal of Pattern Recognition and Artificial Intelligence, vol. 18, no. 3, pp. 475–496, 2004.

30. K. Sambhoos, et al., “Enhancements to high level data fusion using graph matching and state space search”, Submitted to IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, 2006.

3 BACKGROUND

In current and future operational environments such as Global War on Terrorism (GWOT) and Maritime Domain Awareness (MDA), warfighters require technologies evolved to support information needs regardless of location and consistent with the user's level of command or responsibility and operational situation. To support this need, the DOD has developed the concept of Network Centric Warfare (NCW). The Chief of Naval Research (CNO) defines NCW as "military operations that exploit state-of-the-art information and networking technology to integrate widely dispersed human decision makers, situational and targeting sensors, and forces and the contractor aprons into a highly adaptive, comprehensive system to achieve unprecedented mission effectiveness." Net-centric operations include communications and information assurance capabilities to enable all-source data access, multi-source processing, and tailored dissemination to Command and Control (C2) and Intelligence, Surveillance, Reconnaissance (ISR) users across the network.

The operational benefits sought are an increased speed, accuracy and precision of command; distributed self-synchronization; flexibility and adaptability to an operational situation; and decision superiority. To accomplish this, it must be possible to automate understanding of the battlespace by identifying objects, determining relationships among the objects, assessing intent, and automatically generating courses of action with associated risks and uncertainty.

To meet the operational benefits described above, the H^2LIFT project was contracted to advance the progression of Level 2/3 fusion of informational content to obtain an advanced multi-intelligent system for hierarchical high level decision making processes. This objective was met with the development of an information integration mechanism to simplify human decision making solving operational problems. As technology continues to advance, and the proliferation of sensors in all platform increases, the human decision makers are being overwhelmed with data. In this research, the CUBRC developed a novel approach in the near "real-time" detection of hypotheses in asymmetric warfare scenarios (i.e. urban warfare, maritime domain awareness).

4 SCOPE

The primary objective of this effort was the progression of Level 2/3 fusion of informational content to obtain an advanced multi-intelligent system for hierarchical high-level decision making processes. The goal was to develop an information integration mechanism to simplify human decision making solving operational problems. As technology continues to advance and the proliferation of sensors in all platform increases, human decision makers are being overwhelmed with data. In this research, the CUBRC proposed a cost effective two-year program of a novel approach in the near "real-time" ranking/formulation of hypotheses in asymmetric warfare scenarios. In particular, CUBRC introduced the Hierarchical High Level Information Fusion Technologies (H^2LIFT) architecture with the following objectives:

- develop H^2LIFT Architecture and algorithms for GWOT/MDA threats,
- develop prototype software that implements H^2LIFT architecture and algorithms, and
- develop a simulation based tool for performance evaluation and analysis

Since all required Level 1 type sensor information was available a priori, the primary focus of the research was Level 2/3 fusion (Situation and Threat Assessment). The hypotheses were tested using a developed simulation software package with predefined evaluation metrics. The evaluation metrics included Level 2/3 fusion assessment tools using a realistic naval threat scenario example. The software model simulated a naval threat from an incoming vessel (such as a cargo ship containing a weapon of mass destruction), included in a group of non-threatening vessels, to access the Level 2/3 H^2LIFT algorithm. The simulation package was used as an evaluation measure and performance platform providing an operational utility assessment tool. Although sensor data was assumed to be available, the simulation package had the capability of delaying sensor information in time to model the lag in available sensor data.

5 SOW STATUS

The following paragraphs describe the outcomes of the requirements and deliverables identified in the Statement of Work (SOW), which was included in the referenced contract as Attachment 1.

5.1 Requirements

5.1.1 Develop H^{2}LIFT Architecture and Algorithms for GWOT/MDA Threats

Requirement Text: The Contractor shall leverage existing Graph Theoretical approaches to demonstrate an alternative to using pure probabilistic (i.e. Bayesian Networks) or pure rule-based (i.e. Expert Systems) in the aggregation process of Level 2/3 fusion. The IF architecture will define the tactical and strategic decisions needed to create a more efficient human-in-the-loop and to resolve data inconsistencies that are common in a distributed data fusion process.

5.1.1.1 High Level Architecture

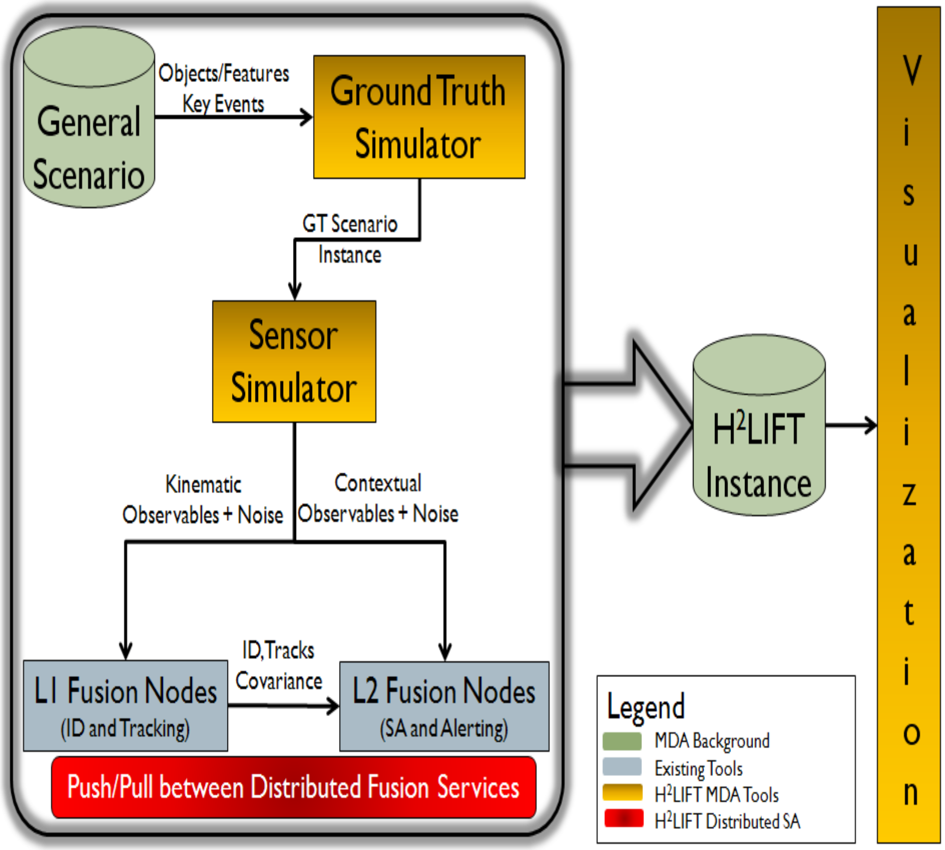


Figure 1 - H2LIFT High Level Architecture

5.1.1.1.1 Introduction To The H^{2}LIFT Project

The H^{2}LIFT project focuses on the progression of information from Level 2 to Level 3 fusion obtaining an advanced multi-intelligent system for hierarchical high level decision making processes. As technology advances and the number of sensors in all platforms increases, the human decision makers are being overwhelmed with data. This leads to slowed response times as well as misguided decisions and overlooked warnings. The goal of the H^2LIFT project is to provide a unique means of filtering superfluous information among the three stages of high level fusion (local, distributed and network centric) so that the amount of data transferred is minimized but remains complete and comprehensive. Therefore, the goal is to provide a mechanism for human decision makers in developing a quick, well-informed decision prompting action while a threat is still in route to the United States. A maritime scenario for testing the effectiveness of the H^2LIFT algorithm is developed with real-world examples of ships carrying cargo to and from the United States as well as sensors for detection of WMD, tracking of ships, etc...

5.1.1.2 General Scenario

The H2LIFT Scenario contains information and raw data for thousands of ships. The amount of ships used is large so it will mimic the amount of data that a real time decision maker has to deal with. There are six situations and six ships that will appear in this scenario. All other ships that appear in the scenario are considered background noise. The ships are there and reporting information but there is not a significant accumulation of corresponding anomalies which would flag the ship as a potential threat.

The following situations are representative examples of possible definitions of evidence supporting a situational estimate. In our definition, anomalies are features of one or more situation classes as described below. The hierarchical fusion process is designed to redefine or add new situations as needed and appropriate. As situation instances accumulate evidence, the credibility of their occurrence increases proportionally and will result in its' appropriate ranking with other estimated occurrences.

Ships 8148 and 6257 are involved in a Contraband Smuggling situation. For ship 8148 a Contraband Smuggling situation was formed because the ship illustrated the following anomalies: Plan Incomplete; Pos Deviation; Unplanned Port Activity. For ship 6257 a Contraband Smuggling situation was formed because of these anomalies: Neg Deviation; Unplanned Port Activity; Pos Deviation.

Ships 5809 and 5602 demonstrate an Illicit Weapon Shipment situation. An Illicit Weapon Shipment situation was formed because the two ships had the following anomalies: Plan Incomplete; AIS Turned Off; Duplicate Container Number.

Ship 3223 demonstrates an Illicit Nuclear Shipment situation. This situation was reported because the ships actions demonstrated the following anomalies: Plan Incomplete; Radio Active Cargo; Seven Day Rule.

Ship 6257 demonstrates a Hijacking and an Attack on US Forces situation. Hijacking and Attack on US Forces were reported because of the following anomalies: Neg Deviation; Pos Deviation; Duplicate Container Number; AIS Swap.

Ship 631 demonstrates a Shipment of Nuclear Material to the US situation. A Shipment of Nuclear Material was formed because of the following anomalies: Plan Incomplete; AIS Turned Off; Seven Day Rule.

Annotated Description

On day 13, ANSHUN (Ship 5809) departs Baoshan CH. The ship is expected to arrive in Kemaman MY. ANL EMBLEM (Ship 5602) departs Baoshan CH. It is expected to arrive in Kemaman MY.

On day 14, APL VIRGINIA (Ship 6257) reports that it is departing Bombay IN and has plans to arrive at Haeju KN.

On day 15, ADELHEID S (Ship 631) reports that it is departing Baoshan CH and will be arriving at Kemaman MY.

On day 19, ASTRAVALENTINA (Ship 8148) departs Galveston US. The report that was received and the ships plan did not specify a destination port.

On day 20, ANSHUN departs Kemaman MY and is expected to arrive back in Baoshan CH. ANL EMBLEM departs Kemaman MY and is expected to arrive back in Baoshan CH.

On day 21, ALIANCA ANDES (Ship 3223) arrives in Baoshan CH. The ships movement report was incomplete and there was no port of departure listed. ADELHEID S reports that it arrived in Kemaman MY.

On day 22, ALIANCA ANDES was reported to be departing Baoshan CH. The destination port is unknown. ADELHEID S reports that it is departing Kemaman MY and will be arriving at Baoshan CH.

On day 26, ALIANCA ANDES arrives back in Baoshan CH. The origin port of the ship was not reported and is unknown. Container ID NEPU 9870210 is off-loaded from the ship. This container is said to be carrying Radio Active Cargo.

On day 27, ANSHUN departs Baoshan CH and is expected to arrive in Kemaman MY. ANL EMBLEM departs Baoshan CH and is expected to arrive in Nampo KN. The ship never reports that it arrived in Nampo KN.

On day 28, ALIANCA ANDES departs Baoshan CH with a planned destination of Yokkaichi JA. ADELHEID S arrives at Baoshan CH.

On day 29, Container ID NEPU 9870210 is loaded onto ADELHEID S. Container ZEPU 9870210 was offloaded from ALIANCA ANDES on day 26. The container was not at port for 7 days therefore breaking the Seven Day Rule. ADELHEID S reports that it is departing Baoshan CH. The plan is incomplete because the destination port is unknown. Throughout its travels, it is determined that the ships AIS is no longer reporting.

On day 31, ANL EMBLEM arrives back in Baoshan CH. The ship was coming from an unidentified port. ALIANCA ANDES arrives at Yokkaichi JA.

On day 32, ALIANCA ANDES reports that it is departing Yokkaichi JA and will be arriving at Bonaberi CM. APL VIRGINIA is expected to arrive at Haeju KN. A port activity stating the ship has arrived was never reported.

On day 33, ANL EMBLEM loads container ZIMU 99960001 onto the ship. The ship departs Baoshan CH with an unspecified destination. There is no report sent when the ship reaches its destination. It is determined that ANL EMBLEM has its AIS turned off.

On day 34, ANSHUN departs Kemaman MY and is traveling back to Baoshan CH.

On day 35, APL VIRGINIA arrives at Tsingtao CH. The ship came from Haeju KN but departure was not reported.

On day 36, Container ID ACLU 1059490 was loaded onto ship APL VIRGINIA. That container was already reported being loaded onto ship 7839. APL VIRGINIA reports that it is departing Tsingtao CH and will be arriving in Haeju KN. There is no report stating that the ship arrives at Haeju KN.

On day 37, ASTRAVALENTINA departs Lagos NI and is scheduled to arrive at Hong Kong CH. The ship is sensed making a stop at Bonaberi CM on the way to Hong Kong. This stop was not specified in the pre-determined travel plan. Due to the unscheduled stop at Bonaberi, the ship is deviating from its route. ANL EMBLEM arrives in Baoshan CH. The ships origin port is unknown.

On day 39, ASTRAVALENTINA reports that it is leaving Bonaberi CM. ANL EMBLEM is departing Baoshan CH with an expected destination of Yakkaichi JA. APL VIRGINIA is expected to depart Haeju KN with a plan to arrive at Bombay IN. There are no Port Activities reported for departure of port or arrival at port.

On day 40, ANSHUN loads container ZIMU 9996001 onto the ship. ANSHUN departs Baoshan CH but does not specify a destination. Container ZIMU 9996001 is already on ANL EMBLEM. During ANSHUN's travels, their AIS is no longer reporting.

On day 43, ANL EMBLEM is departing Yokkaichi JA and is expected to arrive in Bonaberi CM.

On day 46, ADELHEID S is expected to depart Bandar Taheri IR and will arrive at Baoshan CH.

On day 54, ANSHUN reports that it is planning to depart Bandar Taheri IR and travel to Baoshan CH.

On day 55, APL VIRGINIA is expected to depart Bombay IN with a plan to arrive at Hong Kong CH.

On day 57, APL VIRGINIA was reported arriving in San Francisco US. APL VIRGINIA was expected to arrive at Hong Kong CH. This stop at San Francisco is unexpected. This unexpected stop caused a deviation to occur because it doesn't match the expected route for the ship. It was also determined that the rigging code for ship 6257 was changed resulting in an AIS swap.

On day 63, ADELHEID S is expected to depart Baoshan CH and will arrive in Kemaman MY.

5.1.1.3 Ground Truth and Sensor Simulator

5.1.1.3.1 Introduction

This simulation supports a fusion algorithm (H^2LIFT) for collecting and analyzing data providing a heuristic analysis tool for detecting weapons of mass destruction in the maritime domain. Tools required to develop a navigational simulation fitting a set of project objectives are introduced for integration into the H^2LIFT algorithm. Emphasis is placed on the specific requirements of the H^2LIFT project, however the basic equations, algorithms, and methodologies can be used as tools in a variety of scenario simulations. Discussion will be focused on track modeling (e.g. position tracking of ships), navigational techniques, WMD detection, and simulation of these models using Matlab and Simulink. Initial results provide absolute ship position data for a given multi-ship maritime scenario with random generation of a given ship containing a WMD. Required coordinate systems, conversions between coordinate systems, Earth modeling techniques, and navigational conventions and techniques are introduced for development of the simulations.

5.1.1.3.2 Conversion Factors

In this section unit conversions used in the development of the simulation algorithm are presented:

- 1 knot = 1 nautical mile per hour = 1.151 statute miles per hour.
- 1 nautical mile = 1.151 statute miles = 6076.115 feet = length of 1 minute of longitude at the equator = 10 cables length = 1000 Fathoms = 0.333 Leagues.
- 1 Fathom = 6 feet
- 100 Fathoms = 1 Cable length

5.1.1.3.3 Reference Frames

In this section reference frames for Global Positioning System (GPS) and Inertial Navigation System (INS) applications are presented:

- Earth-Centered-Inertial (ECI): A right-handed coordinate system denoted by $\{\hat{\mathbf{i}}_1,\hat{\mathbf{i}}_2,\hat{\mathbf{i}}_3\}$. The $\hat{\mathbf{i}}_1$ axis points towards the vernal equinox, the $\hat{\mathbf{i}}_3$ axis points toward the North pole, and the $\hat{\mathbf{i}}_2$ axis follows the rules of a right handed coordinate system. Note that the ECI frame does not rotate with respect to the stars (except for procession of equinoxes) and the Earth turns relative to this frame. Vectors described using ECI coordinates will have a superscript *I.* [1]

- Earth-Centered-Earth-Fixed (ECEF): A right-handed coordinate system denoted by $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$. This frame is similar to the ECI frame with $\hat{\mathbf{e}}_3 = \hat{\mathbf{i}}_3$; however, the $\hat{\mathbf{e}}_1$ axis points in the direction of the Earth's prime meridian, and the $\hat{\mathbf{e}}_2$ axis completes the right hand coordinate system. Note that the ECEF frame does rotate with the Earth. Vectors described using ECEF coordinates will have a superscript *E.* [1]

- North-East-Down (NED): A right-handed coordinate system denoted by $\{\mathbf{n},\mathbf{e},\mathbf{d}\}$. The **n** axis points directly north, the **e** axis points directly east, and the **d** axis completes the right-handed coordinate system (pointing towards the center of the earth on a spherical earth model). Note that the plane formed by the **n** and **e** axes is tangent to the Earths surface with the **d** axis perpendicular to the Earths surface. This makes it convenient to use for local navigation purposes. Vectors described using NED coordinates will have a superscript *N.*[1]

- Gravity Axes: A right-handed coordinate system denoted by $\{\boldsymbol{\Xi}_g,\boldsymbol{\Psi}_g,\mathbf{Z}_g\}$. The $\mathbf{Z}_g$ axis points towards the Earths center, $\boldsymbol{\Psi}_g$ points towards the south pole, and $\boldsymbol{\Xi}_g$ completes the right-hand system by pointing East. Note that like the NED coordinates, this coordinate system is based at the Earths surface.

5.1.1.3.4 Modeling The Earth

Three basic types of Earth models are commonly used for describing motion relative to the fixed Earth frame differing in their assumption of the Earths shape. This section describes each Earth model in order of increasing accuracy, as well as modeling techniques used for the H^2LIFT simulations.

- **Flat Earth:** These models are used for plane surveying over distances short enough so that the Earths curvature is insignificant (less than 10 km).[2] These models get increasingly less accurate with increasing distances spanned and therefore are not a practical tool to use for global navigation applications.

- **Spherical Earth:** These models represent the shape of the earth as a sphere of constant radius. Spherical models fail to model the actual shape of the earth, but are surprisingly accurate for distances spanned near the equator, short-range navigation, and global distance approximations.[2] These models get increasingly less accurate towards the poles of the Earth as a result of its failure to represent the Earths true shape. The slight flattening of the Earth at the poles results in about a 20 km difference between the average spherical radius and the measured polar radius of the earth. [2]

- **Ellipsoidal Earth:** These models are required for accurate range and bearing calculations over long distances. Loran-C and GPS navigation receivers use ellipsoidal earth models to compute position and waypoint information. These models define an ellipsoid with an

> equatorial radius and a polar radius. The best of these models can represent the shape of the earth over the smoothed, average sea-surface to within about one hundred meters. [2]

The H^2LIFT project is focused on the progression of information through the fusion levels and is not particularly concerned with the source of this information. The sensor data generated from the scenario simulations should be realistic however the project is predominately concerned with the processing of this information rather than the development of it. Consequently, a Spherical Earth Model will be sufficiently accurate and convenient to implement as the groundwork for H^2LIFT scenario simulations.

5.1.1.3.5 Geodetic vs. Geocentric Latitude

The angle L' in figure 1 is called "geocentric latitude" and is defined as the angle between the equatorial plane and the radius from the Earths geocenter. The angle L is called "geodetic latitude" and is defined as the angle between the equatorial plane and the normal to the surface of the ellipsoid. The word "latitude" typically refers to geodetic latitude, which is the basis for most of the maps and charts we use.

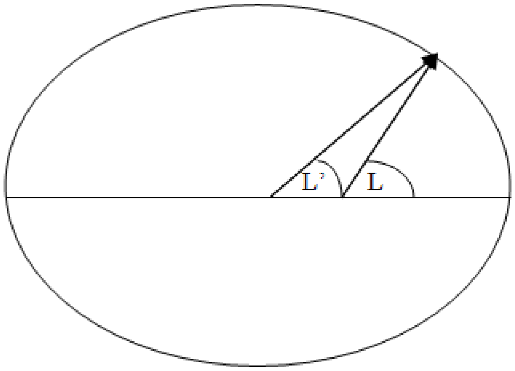


Figure 2 - Difference between Geodic and Geocentric Latitudes

Note, there is no distinction between geocentric and geodetic longitude, as longitude is always taken as geodetic. Also, for a spherical earth model the geocentric latitude becomes the geodetic latitude and thus is not an issue for the H^2LIFT project.

5.1.1.3.6 Symbols Used

Ξ = Velocity of Vehicle with respect to X- gravity axis.

Ψ = Velocity of Vehicle with respect to Y- gravity axis.

Z = Velocity of Vehicle with respect to Z- gravity axis.

λ = Geocentric Longitude.

L = Geocentric Latitude.

h = altitude

γ = Flight Path Angle.

H = Heading Angle (True Course), (+ CW from N).

ς_e = Rate of rotation of the Earth.

R_e = Spherical Radius of the Earth.

χ = Euler angle defining the alignment of the Vehicles reference X-axis with respect to the Gravity X-axis.

Θ = Euler angle defining the alignment of the Vehicles reference Y-axis with respect to the Gravity Y-axis.

ϕ = Euler angle defining the alignment of the Vehicles reference Z-axis with respect to the Gravity Z-axis.

5.1.1.3.7 H^{2}LIFT Earth Model Assumptions.

- Spherical Earth
- Ships do not leave the Earth's surface. (h = h = $\gamma = \gamma = 0$. And the Earths rotation can be neglected in the equations since the ships will be rotating with the Earth as opposed to the Earth rotating with respect to an airborne vehicle.)
- Since the ships travel at the Earths surface, their vehicle reference axes will be based on the Earths surface along with the gravity axes. For simplicity, each vehicle reference axes system is fixed corresponding with the gravity axes system ($\chi = \Theta = \phi = 0$).

5.1.1.3.8 Resulting H^{2}LIFT Earth Model Equations (Taken from Reference 3)

- $$\frac{d\gamma}{dt} = \tan^{-1}\left\{ \frac{\frac{dh}{dt}}{\sqrt{\left(\frac{dX}{dt} - R_e\Omega_e \cos(\ )\right)^2 + \frac{dY}{dt}^2}} \right\} = 0 \qquad \textbf{(1)}$$

- $$\frac{d\lambda}{dt} = \frac{\frac{dX}{dt}}{R_e \cos(\ )} - \Omega_e = \frac{\frac{dX}{dt}}{R_e \cos(\ )} \qquad \textbf{(2)}$$

- $$\frac{dL}{dt} = \frac{-\frac{dY}{dt}}{R_e} \qquad \textbf{(3)}$$

5.1.1.3.9 Navigating The Earth

This section introduces the method of great circle navigation as well as details of rhumb line navigation. Rhumb line navigation is the route calculation method of choice for the H^{2}LIFT project.

5.1.1.3.10 Great Circle Navigation

The shortest distance between two points on the Earths surface is a straight line that lies entirely inside of the Earths surface. This is not a realistic option when deciding transportation routes. The shortest distance following the Earths surface (i.e. that can be realistically traversed) actually lies vertically above the aforementioned straight-line route. This idea gives rise to the concept of great circle navigation. Conceptually, the construction of a great circle route involves creating an imaginary plane that intersects the two points of interest as well as the center of the earth. This plane will slice the (assumed spherical) earth into 2 hemispheres of equal size. The circumference of the flat face of each hemisphere is then called a great circle. Note that only planes that cut through the center of the earth give rise to great circles. For instance, all meridians are great circles, while the equator is the only line of latitude that is a great circle.[4]

5.1.1.3.11 Rhumb Line Navigation

A shortcoming of great circle navigation is that the heading angle varies continuously. For example, the great circle route between two points of equal (non-zero) latitude does not follow the line of latitude in an E-W direction, but arcs towards the pole. However, it is possible to traverse two points using a constant heading angle by using a method called rhumb line navigation.[4] It is because of this fact that rhumb line navigation was chosen as the basis for the H^2LIFT simulation, as it would be significantly easier to program the heading angle to remain constant between waypoints than it would be to vary it continuously. Notice that a pilot flying a great circle route for a sufficient amount of time would encircle the earth and end up back where he started, while a pilot flying a rhumb line route would spiral indefinitely poleward.[4]

Since the H^2LIFT project is focused on ships rather than airborne vessels, each shipping route in its entirety will be made up of a series of short rhumb line segments between consecutive waypoints in order to avoid landmasses. Since each individual rhumb line segment will be short, the difference in route length as compared to using great circle navigation will be minimal while the advantage of programming simplicity will be considerable.

Rhumb lines satisfy the following equation taken from reference 5:

$$H = \frac{2(\lambda_2 - \lambda_1)}{\log\left\{\left[\frac{(1+\sin(\phi_2))}{(1-\sin(\phi_2))}\right]\times\left[\frac{(1-e\times\sin(\phi_2))}{(1+e\times\sin(\phi_2))}\right]^e\right\} - \log\left\{\left[\frac{(1+\sin(\phi_1))}{(1-\sin(\phi_1))}\right]\times\left[\frac{(1-e\times\sin(\phi_1))}{(1+e\times\sin(\phi_1))}\right]^e\right\}}$$

***With a spherical earth (e = 1) and the equation becomes:

- $$H = \frac{2(\lambda_2 - \lambda_1)}{\log\left\{\left[\frac{(1+\sin(\phi_2))}{(1-\sin(\phi_2))}\right]\times\left[\frac{(1-\sin(\phi_2))}{(1+\sin(\phi_2))}\right]\right\} - \log\left\{\left[\frac{(1+\sin(\phi_1))}{(1-\sin(\phi_1))}\right]\times\left[\frac{(1-\sin(\phi_1))}{(1+\sin(\phi_1))}\right]\right\}} \qquad \textbf{(4)}$$

5.1.1.3.12 Simulation Specifics

This section will present the methodology and techniques used to create the H^2LIFT simulation. It is broken up into sections that correspond to each of the main subsystems in the following diagram.

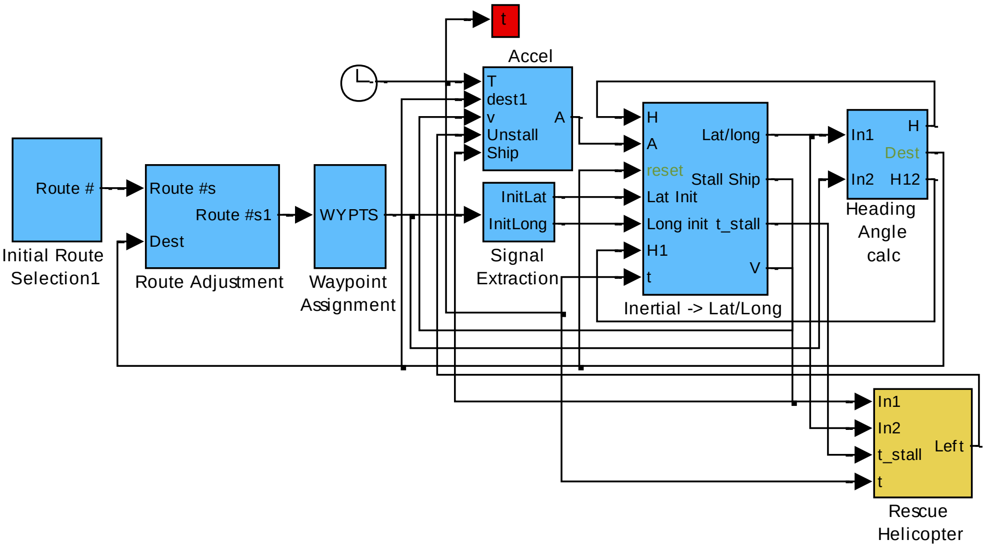


Figure 3 - Kinematic Simulation Architecture

5.1.1.3.13 Initial Route Selection

This subsystem generates a variable sized vector of random integers, each corresponding to an individual ships route number and sorts them in ascending order. The variable used for the H^2LIFT project is ‘ShipQuantity’ and is defined in an initialization m-file that is run prior to the Simulink model. The random numbers are generated with a constant block equipped with the following argument:

- Floor(rand(1,ShipQuantity)*16.9) **(5)**

This argument creates a constant vector of length ‘ShipQuantity’ of integers between zero and 16 and corresponds to the initial route numbers for each of the ships.

5.1.1.3.14 Route Adjustment

This subsystem randomly selects a variable number of ships to contain WMD’s and ensures that these ships are assigned routes destined for the United States. It also increments a particular ships route number when it completes its previous route. This variable is assigned in the initialization m-file and is referred to as ‘NumWMD’ in the H^2LIFT model. The route numbers are set up such that the destination port of a particular route is also the departure port of the following route. This way, incrementing the ship routes by 1 each time will avoid any jump discontinuities in a particular ships position when transitioning routes. The final output of the subsystem is a vector of route numbers in ascending order, with at least ‘NumWMD’ quantity of them destined for the United States.

5.1.1.3.15 Waypoint Assignment

This subsystem takes the vector outputted from the ‘Route Adjustment’ subsystem and assigns the corresponding sequence of waypoints to the particular route numbers. It is vital to ensure that the output signal is organized appropriately so that it can be used by subsequent subsystems (namely, when it is de-muxed in the ‘Heading Angle Calculation’ subsystem it needs to organized a specific way in order to be de-muxed appropriately). This is very tricky considering that the number of ships, and hence the size of the signal is a variable. This order is as follows:

Ship 1 Waypoint 1 Latitude

Ship 2 Waypoint 1 Latitude

.

.

.

Ship n Waypoint 1 Latitude

Ship 1 Waypoint 1 Longitude

Ship 2 Waypoint 1 Longitude

.

.

.

Ship n Waypoint 1 Longitude

Ship 1 Waypoint m Latitude

Ship 2 Waypoint m Latitude

.

.

.

Ship n Waypoint m Latitude

Ship 1 Waypoint m Longitude

Ship 2 Waypoint m Longitude

.

.

.

Ship n Waypoint m Longitude

5.1.1.3.16 Signal Extraction

This subsystem extracts all of latitude and longitude coordinates from the vector created by the ‘Waypoint Assignment’ subsystem to be used as initial conditions for the position integrator blocks in the ‘Convert Inertial measurements to Latitude and Longitude Coordinates’ subsystem. This is also challenging since the number of values to extract is dependent upon a variable quantity (namely the number of ships). This is another instance in which it is important to know the organization of the input signal to ensure that the appropriate values are being extracted.

5.1.1.3.17 Acceleration

This subsystem generates a signal corresponding to the ships acceleration. The signal will be equal to the magnitude of acceleration for the amount of time required to reach the desired ship velocity and zero at all other times. The subsystem is capable of decelerating a random ship at a random time to reach a slower velocity and reaccelerating it to its top speed at a random future time. It is also capable of accelerating the ships after they are repaired if they stall, and when they depart ports. Most of the synchronization of the simulation scenarios is done with this subsystem by altering the departure times and port delays. Top velocities and accelerations should be consistent with the type of ship being accelerated.

5.1.1.3.18 Convert Inertial Measurements to Latitude and Longitude Coordinates

This subsystem takes the acceleration signals generated by the ‘Acceleration’ subsystem and integrates them to obtain velocity signals. These velocities are then broken up into their Northern and Eastern components using the Heading angle calculated in the ‘Heading Angle Calculation’ subsystem and used in the equations **2** and **3** above to find differentials of latitude and longitude changes. These differentials are then integrated with the initial conditions set to the departure port coordinates of the current route to find the current latitude and longitude coordinates for each ship. Each time a ship reaches its destination port for a given route or is randomly chosen to break down, the velocity integrator must be reset to zero. Also, each time a ship reaches its destination port the position integrators must be reset to an external initial condition of the new routes departure port coordinates.

5.1.1.3.19 Heading Angle Calculation

This subsystem compares the current location of the ship to the various waypoint coordinates and then calculates the heading angle between the ships current location and the next successive waypoint using equation **4** above. The heading angle is continuously being updated as the ships current position changes. To avoid sharp angled turns at waypoints the heading angle signal is conditioned with the following transfer function that creates a gradual transition between waypoints.

- $$\frac{1}{\tau S + 1} \qquad \textbf{(6)}$$

Where tau is the time constant of the gradual change. This transfer function must be converted into a differential equation in order to support multiple ships heading angles simultaneously. While using this gradual heading angle transfer function it is important to precondition the heading angle to be sure that it is positive. For example, any negative heading angle should have 2Π added to it to convert it to its positive value. This will avoid any loops in the ship track while gradually changing the heading angle.

5.1.1.3.20 Rescue Helicopters and Cargo Trucks

Other vessels such as helicopters and cargo trucks can be added with a very similar methodology. In the case of the H^2LIFT project, the specific vehicle dynamics are not important and thus implementation of other types of vessels can be greatly simplified. A cargo truck for example can be treated exactly as a cargo ship, but with its route entirely on land. The routes should however be consistent with the existing highway system for realistic scenario generation. A rescue helicopter is added by replacing R_e (spherical radius of the earth) in equations **2** and **3** with R_e + h (altitude). This ignores the take off and landing portions and other dynamics of the helicopters flight, but is sufficient for the goals of the H^2LIFT project. The difficult part of the rescue helicopter is that its departure time and destination coordinates depend on when a particular ship stalls and its location when it stalls. Since all of these parameters are randomly selected each time the simulation is run, the helicopters ‘Acceleration’ and ‘Heading Angle Calculation’ subsystems must communicate with the ship model.

5.1.1.3.21 RESULTS

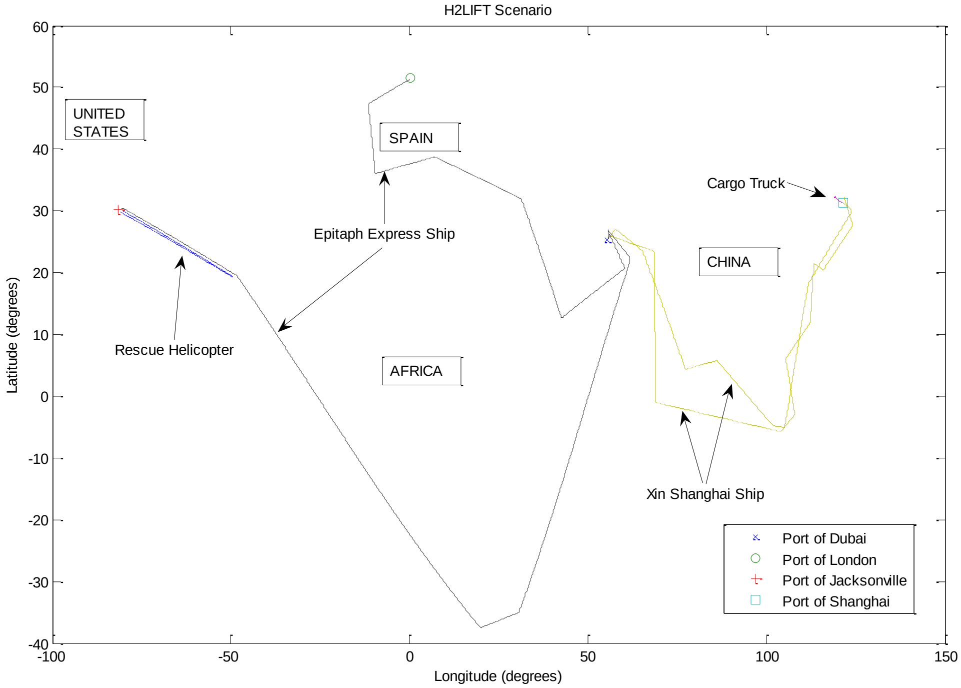


Figure 4 - Plot of Route Tracks of a 2 Ship, 1 Rescue Helicopter, and 1 Cargo Truck Scenario

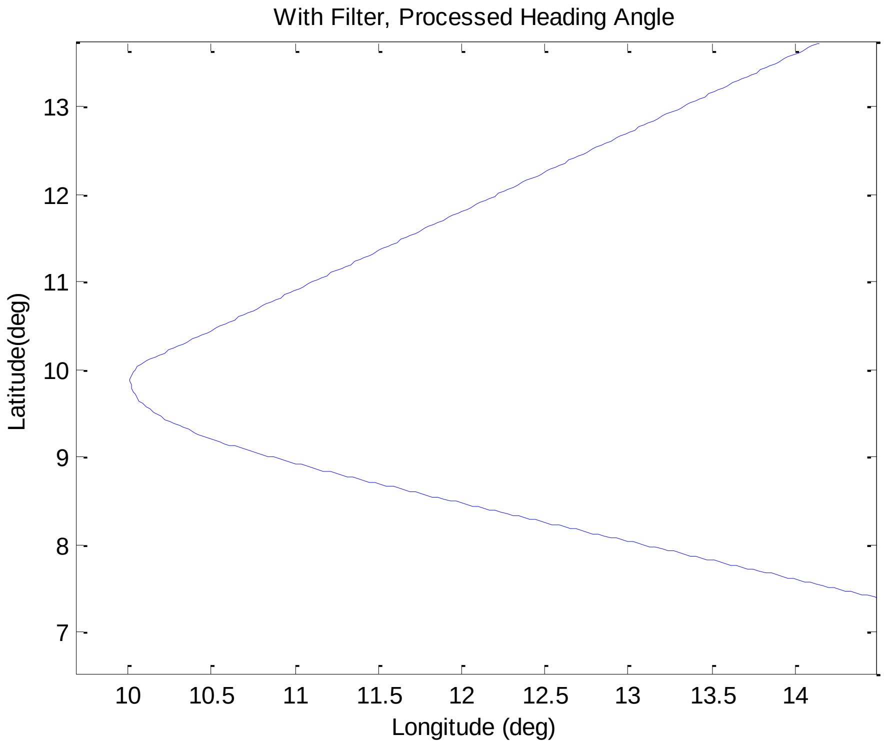


Figure 5 - Close-up of a Gradual Heading Angle Transition. Without use of Eq. 6 this turn would not have a radius.

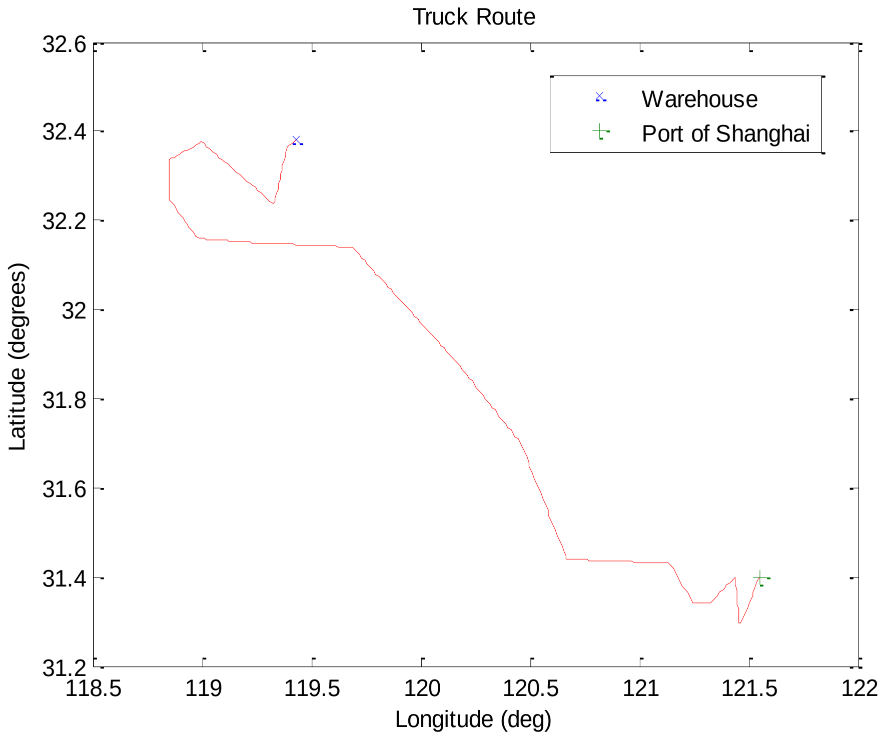


Figure 6 - Close-up of Cargo Truck Route

5.1.1.4 L1 Fusion

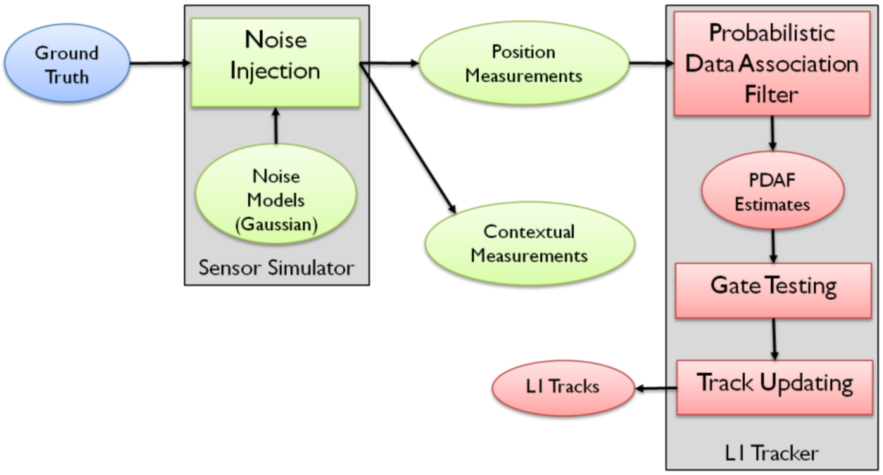


Figure 7 - L1 Fusion Architecture

This section discusses the design of the ship tracker filter. The purpose of the filter is to use realistic measurements of non-littoral ship movements to track a ship and provide probabilistic bounds on the position/velocity errors using 3-sigma ellipses. The tracking filter uses a Kalman-based approach where the continuous-time acceleration is modeled by zero-mean Gaussian white noise process (also known as an α–β filter). It is assumed that the latitude and longitude motions are decoupled, so that a single-axis model is given for each separately. The kinematics model for a single-axis position/velocity state system is given by

$$\begin{bmatrix}\dot{x}\ \ t\\ \ddot{x}\ \ t\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x\ \ t\\ \dot{x}\ \ t\end{bmatrix}+\begin{bmatrix}0\\ 1\end{bmatrix}w\ \ t \tag{1}$$

where $x\ \ t$ and $\dot{x}\ \ t$ are longitude and longitude-rate, respectively, $w\ \ t$ is a zero-mean Gaussian process with spectral density q_x. Note that the first state does not contain any process noise in this formulation. This is due to the fact that this state represents a kinematical relationship that is valid in theory and in the real-world, since velocity is always the derivative of position. Discrete-time measurements of position are assumed, so that

$$\tilde{x}_k = x_k + v_k \tag{2}$$

where the subscript k denotes a quantity at time t_k and v_k is zero-mean Gaussian white-noise process with standard deviation σ_n. It is possible to convert Eq. (1) to discrete time as well. The state transition matrix is given by

$$\Phi=\begin{bmatrix}1 & \Delta t\\ 0 & 1\end{bmatrix} \tag{3}$$

where Δt is the sampling interval. The discrete-time process noise covariance matrix is given by

$$Q_x = q_x\begin{bmatrix}\Delta t^3/3 & \Delta t^2/2\\ \Delta t^2/2 & \Delta t\end{bmatrix} \tag{4}$$

Note the correlations between position and velocity, which can be essentially ignored for small sampling intervals.

Denoting hats for estimates, the discrete-time state estimation equations for the α–β filter are given by

$$\begin{gathered}\hat{x}_k^+ = \hat{x}_k^- + \alpha_x\ \ \tilde{x}_k - \hat{x}_k^-\\ \dot{\hat{x}}_k^+ = \dot{\hat{x}}_k^- + \frac{\beta_x}{\Delta t}\ \ \tilde{x}_k - \hat{x}_k^-\\ \hat{x}_{k+1}^- = \hat{x}_k^+ + \dot{\hat{x}}_k^+\Delta t\\ \dot{\hat{x}}_{k+1}^- = \dot{\hat{x}}_k^+\end{gathered} \tag{5}$$

where the superscripts + and – denote update and propagation, respectively. The gains α_x and β_x are often treated as tuning parameters to enhance the tracking performance. However, conventional wisdom tells us that tuning these gains individually is incorrect. To understand this concept we must remember that the model in Eq. (1) shows a kinematical relationship. If α_x and β_x are chosen separately, then this kinematical relationship can be lost. This means the velocity estimate may not truly be the derivative of the position estimate, even though we know that this relationship is exact. A more true-to-physics approach involves tuning the continuous-time process noise parameter q_x. From Eq. (4) changes in the velocity over the sampling interval are of the order $\sqrt{q_x \Delta t}$, which can be used as a guideline in the choice of q [1]. The complete solution involves the determination of the Kalman gain through the steady-state covariance solution [2].

The solutions for α_x and β_x given q_x are given by first defining the following "tracking index;"

$$S_x = q_x^{1/2} \Delta t^{3/2} / \sigma_n \qquad (6)$$

Next define the following variables:

$$\begin{aligned} \xi_x &= \frac{1}{2}\left[\left(\frac{S_x^2}{2} + \vartheta_x\right) + \sqrt{\left(\frac{S_x^2}{2} + \vartheta_x\right)^2 - 4S_x^2}\right] \\ \vartheta_x &= \left(4S_x^2 + \frac{S_x^4}{12}\right)^{1/2} \end{aligned} \qquad (7)$$

The gains are given by

$$\begin{aligned} \alpha_x &= 1 - \left(\frac{S_x}{\xi_x}\right)^2 \\ \beta_x &= S_x \sqrt{1-\alpha_x} \end{aligned} \qquad (8)$$

Also, the steady-state variances for position and velocity are given by

$$\sigma_{xx}^2 = \sigma_n^2 \left[\left(\frac{\xi_x}{S_x} \right)^2 - 1 \right] 1 - \alpha_x$$

$$\sigma_{\dot{x}\dot{x}}^2 = \left(\frac{\sigma_n}{\Delta t} \right)^2 \left[S_x^2 \left(\frac{1}{2} - \frac{1}{\xi_x} \right) + \xi_x \right] \left(1 - \frac{\beta_x}{\Delta t} \right) \qquad (9)$$

$$\sigma_{\dot{x}x}^2 = \sigma_{x\dot{x}}^2 = \frac{\sigma_n^2 \, \xi_x}{\Delta t}$$

Equation (9) can be used to determine the 3-sigma ellipses of the position and velocity errors. Similar filter equations are given for latitude. Note that the α–β filter can be shown to be stable as long as $q_i, i \equiv x, y$ is greater or equal to zero [2].

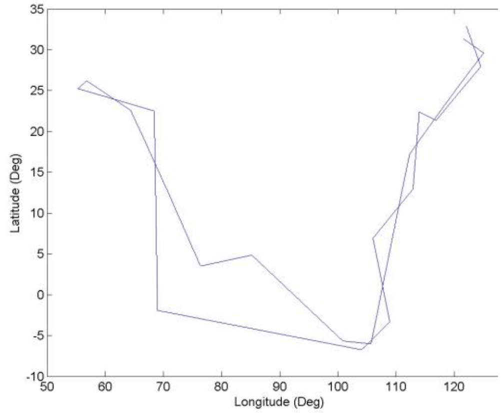


Figure 8 - True Longitude and Latitude

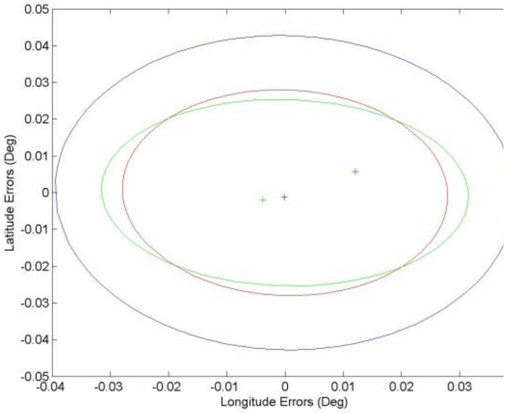


Figure 9 - Selected Error Points and Ellipses

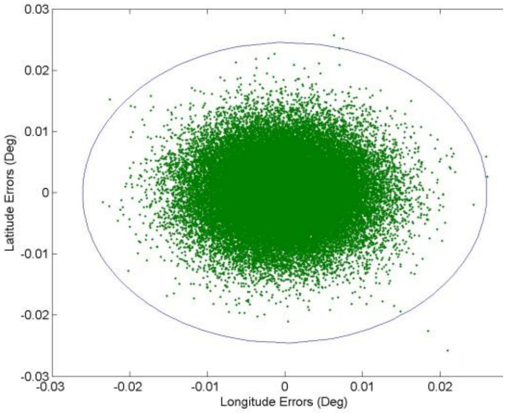


Figure 10 - Error Points and Maximum Ellipse

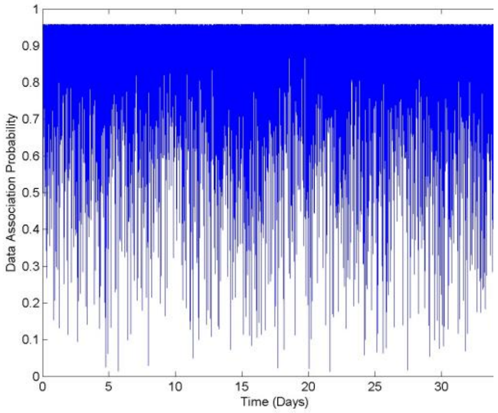


Figure 11 - Association Probabilities

5.1.1.5 L2 Fusion

5.1.1.5.1 Applying L1 Techniques to L2 Fusion

The goal of the L1/L2 research is to apply L1 techniques to L2 information. Here we show an example of tracking an object that is moving. In this case we wish to identify situations of erratic maneuvers or behaviors in a system. The velocity kinematics model is assumed to be given by

$$\dot{v} = -a\,v + w \tag{1}$$

where a is a positive constant and w is zero-mean Gaussian white-noise process with spectral density given by q. Assuming a constant sampling interval, denoted by $\Delta t \equiv t_{k+1} - t_k$, the discrete-time version of Eq. (1) is given by

$$v_{k+1} = e^{-a\,\Delta t} v_k + w_k \tag{2}$$

where the variance of w_k is given by $\frac{q}{2a}\ 1 - e^{-2a\,\Delta t}$. The steady-state value for the variance of v_k, denoted by σ_v^2, is given by solving the classic discrete-time Lyapunov equation [1], which gives a variance of

$$\sigma_v^2 = \frac{q}{2a} \tag{3}$$

Hence a large value of q or a small value of a, i.e. a large time constant in the system, can lead to a large velocity variance and thus “more” erratic behavior in the position. To see how position is affected by a and q, the following model is used

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \equiv F \begin{bmatrix} p \\ v \end{bmatrix} + G\,w \tag{4}$$

where p is the position. Note that the bottom equation in Eq. (4) is identical to Eq. (1). The discrete-time equivalent of Eq. (4) is given by

$$\begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \Phi \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \mathbf{w}_k \tag{5}$$

where

$$\Phi = \begin{bmatrix} 1 & \frac{1}{a}\ 1 - e^{-a\,\Delta t} \\ 0 & e^{-a\,\Delta t} \end{bmatrix} \tag{6}$$

and the covariance of $\mathbf{w}_k$, denoted by Q, is given by solving the following integral [1]:

$$Q = q \int_{t_k}^{t_{k+1}} \Phi\ \ \tau\ \ G\,G^T \Phi^T\ \ \tau\ \ d\tau \tag{7}$$

Since $\Delta t \equiv t_{k+1} - t_k$ is a constant, the integral in Eq. (7) can be solved to give

$$Q = q \begin{bmatrix} \frac{1}{a^2}\left[\Delta t + \frac{2}{a}\ \ e^{-a\,\Delta t} - 1\ + \frac{1}{2a}\ 1 - e^{-2a\,\Delta t}\ \ \right] & \frac{1}{2a^2}\ \ e^{-2a\,\Delta t} - 2e^{-a\,\Delta t} + 1 \\ \frac{1}{2a^2}\ \ e^{-2a\,\Delta t} - 2e^{-a\,\Delta t} + 1 & \frac{1}{2a}\ 1 - e^{-2a\,\Delta t} \end{bmatrix} \tag{8}$$

Note that for small values of $a\,\Delta t$, i.e., the sampling rate is within Nyquist's limit, we have $1-e^{-a\,\Delta t} \approx a\,\Delta t$ and $1-e^{-2a\,\Delta t} \approx 2a\,\Delta t$, which gives $Q \approx q\,G\,G^T$.

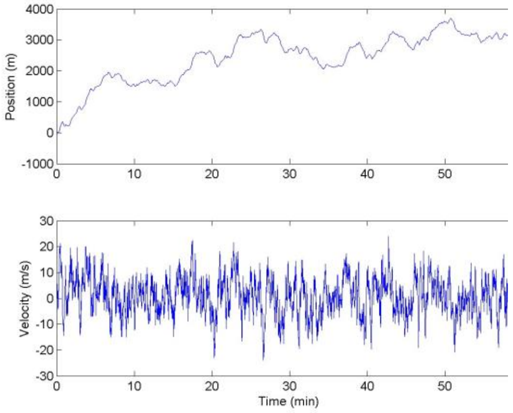


Figure 12 - Position and Velocity with q = $10\mathrm{m}^2/\mathrm{s}^3$

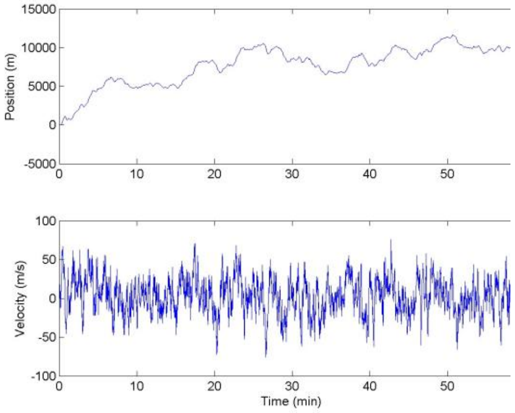


Figure 13 - Position and Velocity with q = $100\mathrm{m}^2/\mathrm{s}^3$

As an example, consider $\Delta t = 1\,\mathrm{s}$ and $a = 0.1\,\mathrm{s}^{-1}$. Results with two difference values for q are given in Figure 12 and Figure 13 (note that the same noise input is used, i.e., the same seed is used to generate the noise). The numerically determined standard deviation of the velocity in Figure 12 is given by 7.2817 and the analytical solution, given by the square root of Eq. (3), is 7.0711, which are in close agreement. Similar results are obtained for Figure 13 (the numerical solution gives a standard deviation of 23.0269 and the analytical solution is 22.3607). Clearly, even though the position patterns are identical, the vehicle in Figure 13 moves a greater distance than the vehicle in Figure 12.

The goal of the research is to use multiple-model adaptive estimation (MMAE) techniques [2-3] to estimate both a and q from noisy position measurements. MMAE techniques employ multiple parallel-running Kalman filters, each using a specific value for a and q, which are weighted using the measurement residual likelihood function. Tables of behaviors will be generated to assess the "erraticness" of the motion. For example, for large values of q/a the motion is highly erratic. Note that the applied approach is general. For example, the same model can also be applied to data sets involving suspicious behavior in individuals. For the vehicle tracking problem other situations can employed though, such as using circular filters as one of the models or using models that confine the motion to a specific path (e.g., the difference between the estimated path and nominally assumed path can be used to assess whether a ship is following a known shipping route).

5.1.1.5.2 Distributed L2 Fusion

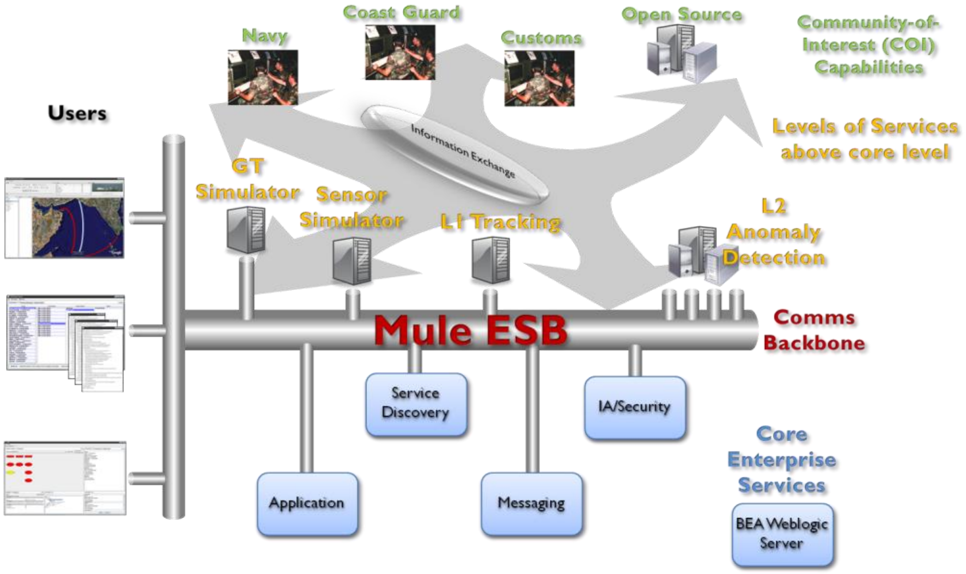


Figure 14 - Distributed L2 Fusion as Services within a SOA

Modern day problem spaces have necessitated a paradigm shift in traditional Information Fusion (IF) systems design. No longer can centralized processing systems ingest the massive amounts of data being generated by the ever increasing number of deployed sensors. No longer can a single system model the entire problem space. Subject matter experts are not skilled in all aspects of the problem and rarely are data sources shared across organizational boundaries. These problem characteristics, among others, justify the Department of Defense's (DoD) transition from stove pipe information systems to network centric Service Orientated Architectures (SOA). This type of architecture is depicted in Figure 14. Some SOA design goals are to achieve ubiquitous IF processing across a globally connected coalition of data providers, data consumers, information analysts, and decision makers. Heterogeneous processing systems can then share information across divisional boundaries, increasing knowledge gain for all decision makers with the fusion of local and global information. Systems operating within the SOA environment are also more flexible to evolving operational roles and data sources. This provides funding agencies with an additional economic incentive to require this type of design. Ubiquitous IF, however, does not come without a price. Increased overhead needed to translate between heterogeneous ontologies, software languages, and protocols consume increased bandwidth which can become a prohibiting factor in bandwidth constrained environments. Informed decisions of which information to share, with what peers, at what time, and how to share it, are questions which must be intelligently answered to succeed in optimizing the use of this bandwidth. Dynamic problem spaces necessitate that these questions be answered by the systems themselves during runtime and in an autonomous fashion.

The maritime domain is defined as all areas and things of, on, under, relating to, adjacent to, or bordering on a sea, ocean, or other navigable waterway, including all maritime-related activities, infrastructure, people, cargo, and vessels and other conveyances [10]. Maritime Domain Awareness (MDA) is the collection, fusion and dissemination of enormous quantities of data — intelligence and information — drawn from U.S. joint forces, U.S. government agencies, international coalition partners and forces, and commercial entities [11]. It has also been defined as the effective understanding of anything associated with the global maritime domain that could impact the security, safety, economy, or environment of the United States [12]. The knowledge gained from this enormous amount of data can then be disseminated to create an operating picture for use by all parties invested in the maritime domain. It is important to note that MDA requires a complete understanding of the activities that are relevant to the maritime environment [13]. Balci and Pegg [13] establish three steps for MDA: i) collecting maritime data, ii) correlating the collected data, and iii) analyzing the collected data, cross-checking and comparing the correlated data from different sources. Note that this operating picture is not necessarily common for all participants. Each participant's realized situational awareness is a function of that participant's decision space. This assertion will form the base motivation for H2LIFT capability, allowing each decision maker access to an amount of information which is necessary and sufficient for the decisions made. The *user* of the proposed fusion services in our research is not a human but a Course Of Action (COA) analysis system to perform resource management and optimization in support of Maritime Interdiction Operations (MIOs). The distributed fusion of disparate heterogeneous data sources within a net-centric environment can improve upon state-of-the-art in the optimal selection of suspect ships for interdiction.

The challenges inherent in solving the MDA problem require the paradigm shift mentioned previously in the introduction to net-centric (distributed fusion) type decision support systems. A fusion system can either be centralized or distributed. Centralized fusion systems are the preferred method due to the optimality of the fused results, but realistic problem constraints rarely allow for this type of system. Instead, distribution of the data, decision makers, and systems across organizational, geospatial, and other spaces require the fusion system to be distributed. Bandwidth constraints restrict the flow of raw data to central locations, and processing constraints restrict the fusion algorithm to run on single machines.

These realistic problem constraints are present in the MDA problem and present rich motivating arguments for the technical approach discussed throughout Section 3. Data and object volumes within the MDA problem are massive, preventing holistic centralization. Not including fishing vessels, yachts, and other smaller unregistered ships, there are approximately 162,000 cargo and 10,000 container registered ships. These ships travel between some 8,000 ports, of which some are cooperative and some are not. In addition there are approximately 200-210 million annual container movements, 13 million of which enter the United States [14]. Automated Information System (AIS) messages alone being generated by these ocean going vessels, total approximately 29.5 million per day. The problem of finding the few pieces of supporting evidence which would indicate high suspicion of a ship within this massive amount of generated data is by no doubt a 'needle in the haystack' problem. Organizations, managing independent sets of resources to collect and analyze information locally relevant to them could feasibly result in knowledge which would assist other organizations. This peer to peer information exchange results in a globally improved situational understanding. How to answer the questions of what information to share, with who, and at what time are the focus of the core research in this paper.

The planning of military COA is an NP-hard problem [15]. In [16], the authors state "tactical event resolution in the manual COA analysis suffers from several problems." The main problems are summarized as:

1. Time constraints only allow for cursory examination of each tactical event.

2. It is difficult to create, share, and maintain common understanding of the actions taken to resolve the tactical events encountered.

3. Opinions among staff officers on the actions which should be taken differ.

4. Staffs have difficulty keeping track of consumed resources.

5. Ordering of engagement within the tactical event is ignored.

COA analysis (i.e., war-gaming) is used to determine the merits of each proposed COAs. The commander selects the 'best' COA to be refined into an operational plan [16]. There have been several papers dealing with automating at least some of the COA analysis. [16] proposes using software agents and genetic algorithms (GA) for event resolution. A multi-objective optimization approach with GAs was proposed in [15]. [17] presented a proof-of-concept prototype to, among other things, utilize intelligent agents to support COA analysis.

Unfortunately, for the global maritime domain, there are large areas without significant sensor coverage. This means that tracking of ships during blue water phases of its voyage may not be feasible. To help alleviate this problem, large commercial vessels that enter U.S. ports are now required to have Automatic Identification System (AIS) capabilities. AIS is a similar system to transponders in commercial aircraft. Every so often, a ships' AIS will broadcast its position and velocity vector. When an AIS receiver is within range of the ship, the ships kinematic progress can be tracked. Currently, AIS receivers are fixed on the earth, but the U.S. Coast Guard has plans to launch a constellation of satellites with global AIS receiving capability [18].

Aerial sensors also have the ability to view and possibly track ship traffic. There are efforts underway to provide near-continuous monitoring of the U.S. and choke-points (e.g., the straits of Malacca), via broad area maritime surveillance (BAMS) UAVs. UAV's outfitted with other sensor platforms such as EO/IR capability can also provide a taskable resource to collect information other than kinematics, e.g., cargo contents, ship type, ship name, etc. Open source websites such as World Shipping Register [19] offer an excellent source of ship characteristics and history. For ships that are insured, the insurance company normally has a voyage plan, detailing the ports of call and estimated times of arrival, the cargo manifest, etc. The capability to fuse this open source data relevant to a ship can greatly increase the effectiveness of higher level fusion services, allowing them to ingest contextual information which otherwise couldn't be sensed. A final important intelligence / data source are human assets. Reports from a human asset at a port can provide information on ships arriving and departing, cargo loaded/unloaded at the port, any crew personnel changes, etc.

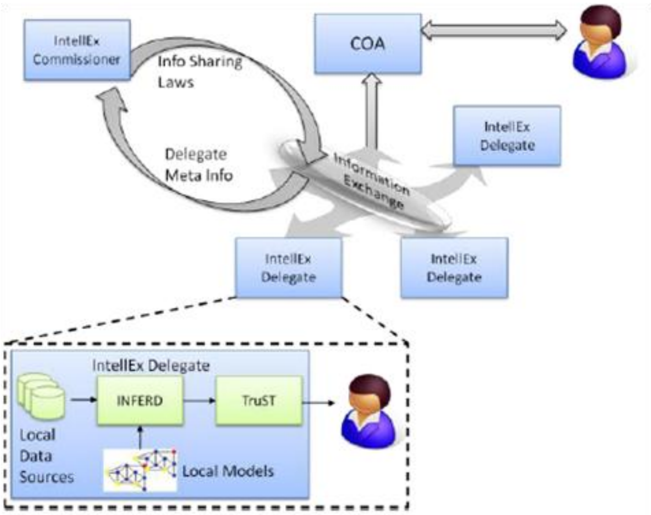


Figure 15 - H2LIFT Fusion Components

The main components of H2LIFT distributed fusion, depicted in Figure 15, are IntellEx, INFERD, TruST, and COA. Each of these sub systems have been developed independently but are being brought together in an attempt to leverage their successes for the achievement of global MDA.

In the figure, we show IntellEx as a distributed information sharing system where a centralized Commissioner generates laws by which peer delegates must obey. The role of each Delegate is to examine their local systems and share information with peer Delegates when within the laws of the Commissioner, and justified by the state estimates of their local hypothesis generation (fusion) systems. INFERD in this case, is the local fusion system to be wrapped by the IntellEx Delegate logic. TruST serves to find the neighboring information within the local INFERD hypotheses which may be relevant for peer fusion services. Finally the global information being shared by the peer Delegation is piped to

COA which generates suggested tasking assignments for a fleet of blue resources. Each of these modules will be discussed in more detail in this section, but the focus will be on the IntellEx distributed information sharing system.

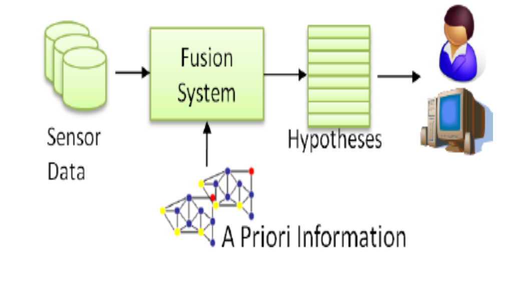


Figure 16 - Generic Local Fusion Process

In Figure 16, we show a generic fusion process. Any fusion system needs at least two things to operate; a priori information and sensor data. In the case of expert systems, this a priori information is encoded explicitly in the form of rules, patterns, Bayesian networks, or others. It is important that this information is encoded explicitly so that the IntellEx Delegate logic can send the necessary information to the IntellEx Commissioner. The other input, Sensor Data, is obvious and needed to drive the hypotheses being generated. Fusion systems exist at many levels as defined by the JDL Model for Data Fusion [20][21] and its recent revision [22]. The level(s) of fusion to most benefit from a robust information sharing system as proposed in this paper would be *high level fusion*. High level fusion, as defined by this model, includes Situation Assessment (L2) and Threat and Impact Assessment (L3). The reason for our proposed system to target high level fusion systems is that there is a greater opportunity to impact the knowledge gain of their users. This is due to the more abstract concepts being hypothesized and the heterogeneous nature of data supporting them. With increasing heterogeneity of the data and magnitude of the models driving the hypotheses, comes an increase of the impact of information sharing on them.

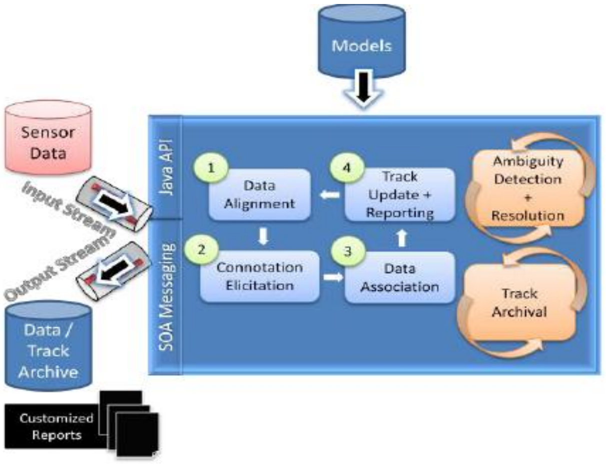


Figure 17 - INFERD High Level Architecture

INFERD [23] was created as a stream based system to update track estimates in real time as sensor messages are produced. We consider INFERD to be L1+, in terms of the JDL data fusion model, due to the measurements which can be applied to the tracks and reports which can be generated across multiple tracks. It is important to note that the tracks being mentioned are not designed to be kinematic based such as those in traditional target tracking; rather they are semantically or contextually based, incorporating a much different type of sensor information. The architecture has been designed in a way such that the models driving the track generation process can make those tracks applicable to many different domains. The inputs to INFERD are twofold: (1) a priori models which define aggregation and elicitation constraints and (2) runtime sensor information which are the observables that are aggregated and elicited from. The output of INFERD is a series of tracks that contain aggregations of meta-tagged observables. Various measurement functions can be loaded to provide a means for quantitative ranking of the tracks and reporting procedures can also be loaded to generate customized output messages for users of the system, be them human or machine.

Figure 17 depicts INFERD's architecture. Within the main box are the processes internal to INFERD, which produces the track set from the a priori models and runtime sensor inputs. The numbered sub modules will be discussed briefly in this paper as they relate to track generation processes found in typical target tracking applications, while the *ambiguity detection* and *track archival* sub modules will be omitted due to scope. The interested reader should consult [23] for details on these two modules. These processes manipulate the track set which is produced, but are not involved in the origination of the tracks.

INFERD currently supports two types of interfacing, also shown in the diagram. The first is the publishing of sensor messages and output messages via SOA messaging interfaces, and the second is an API allowing INFERD to be wrapped within other Java applications.

The four main modules to mention in this paper appear numbered in Figure 17. *Data Alignment* is a process typical to traditional tracking systems which homogenizes the syntax of heterogeneous sensor messages. This provides for common referencing within the fusion processes and makes the process robust to virtually any sensor message type. *Connotation Elicitation* is a unique and important additional stage of processing within INFERD that is not common to traditional target tracking, and which creates value-added for INFERD as a Situational Understanding tool. What necessitates this stage for the tracking of abstract events or entities is the contextual dependency on the meaning of a sensor message. Take for example a sensor message which contains rigging information of a ship. This information would only become suspicious if it was within the context of cargo transport not matching the rigging type. In target tracking, an object is simply a physical target and the tracking is constrained to kinematic features of the terrain, and object. These restrictions can be seen as constraints within a type of inherent model to the tracking algorithm. When attempting to track abstract events and entities such as shipping lane deviations and smuggling, this inherent model must be made explicit to make the same algorithm portable to applications other than the one it was designed under. Hence, necessitating a Connotation Elicitation phase as a precursor to Data Association and Track Update. *Data Association* is a function which has history with multiple target tracking [24]. INFERD tracking shares a similar concept of data association, with a different type of implementation algorithm. In traditional multiple target tracking (MTT), a new measurement received is fused as a position update. This requires association to a track at the track level. This idea is extended within INFERD to accommodate multiple types of association within a single track: (1) new concept and new feature, (2) new feature, existing concept, (3) existing feature, existing concept. This adds complexity to the association process, but provides for a refined association result which was necessitated by the idea of tracking abstract concepts. The inherent kinematic probabilistic constraints used within traditional tracker association algorithms are made explicit in the INFERD engine to accommodate the abstract and contextually dependent types of messages which must be assigned to the tracks. The detailed processing can be found in [23]. The *Track Update* module has the job of taking an associated message and updating the state of its respective track. In the traditional type of kinematic target tracks, every associated message in a track is basically a position and velocity. With this constraint relaxed, tracks now contain contextually dependent heterogeneous message types. For these messages, the update may take on the form of instantiating a new concept within a track, updating an existing concept within a track, or determined to be redundant in which the state is not updated at all. This constraint relaxation, again, makes INFERD more robust to handling non-kinematic type information.

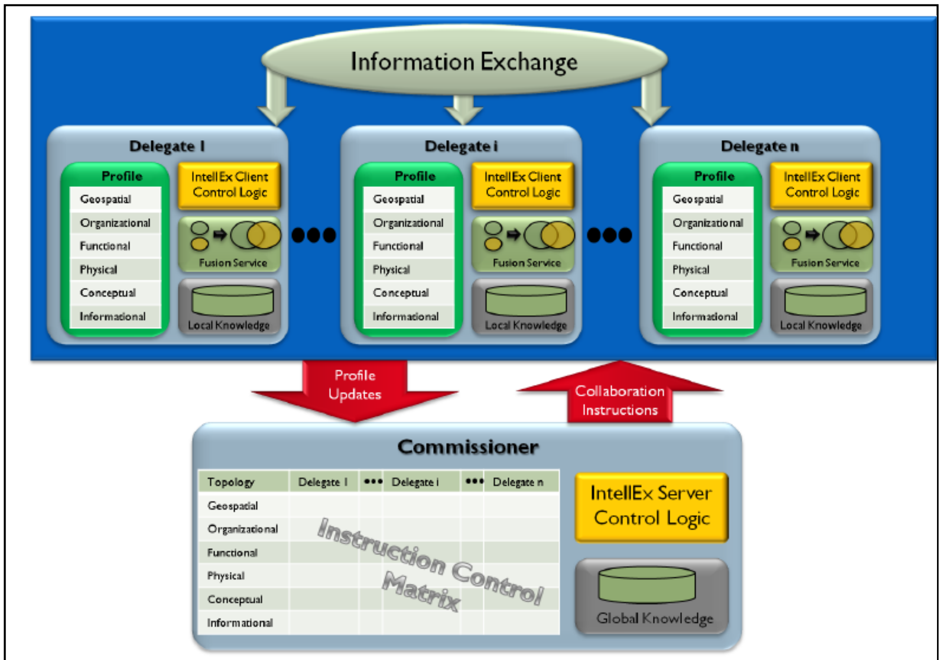


Figure 18 - IntellEx High Level Architecture

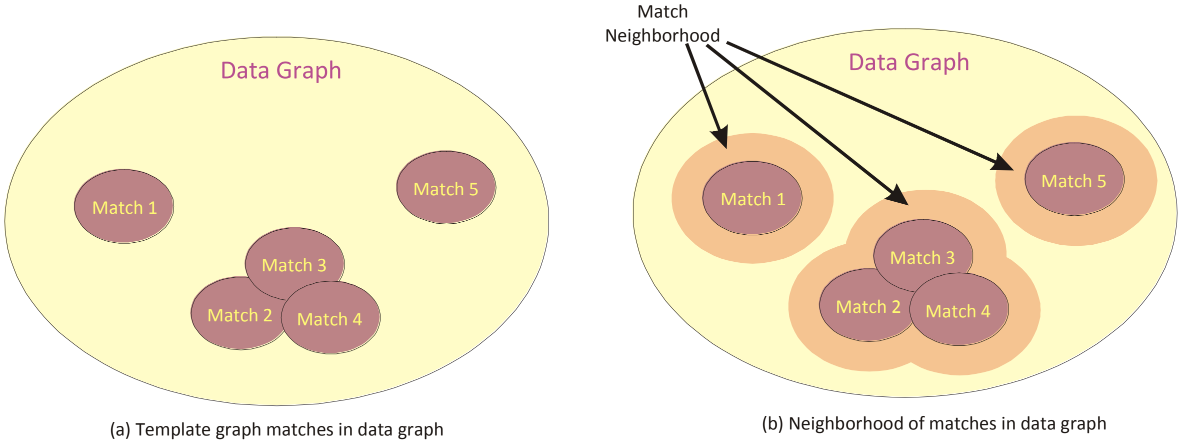
INFERD, in its current instantiation, as well as many other current fusion systems have no means of intelligently sharing their culmination of knowledge with other fusion systems. Moreover, how would a fusion system lacking this capability be able to ingest the information being broadcast by other fusion systems? This is the problem being addressed by the Intelligence Exchange (IntellEx) system (Figure 18). The idea is to create wrappers (termed *Delegates*) which monitor the hypotheses being created and share an extraction of the contained information with other peer Delegates. Figure 5 depicts the high level architecture of IntellEx. The two main components to the architecture are the *Delegates* and the *Commissioner*. Delegates are client fusion service wrappers which enable the communication of hypothesized information within its local fusion service with other peer delegates. The decision process undertaken by each Delegate is governed by the *Collaboration Instructions* provided by the Commissioner.

The challenge faced by IntellEx is a demanding one, and to make measureable success throughout the course of the research, some initial and facilitating assumptions need to be made. One such assumption is that the fusion systems being wrapped are homogenous in terms of both process and ontology. That is for example, graph matching systems are not sharing information with Bayesian systems. To do this would require transformations of the estimates coming out of the systems and the information coming in. While transformations are possible, they would notionally be system dependent requiring transformation modules to be written for different types of fusion systems. This is a complicated issue that will be left for future research.

In the IntellEx system, the sensor data types (not the data itself), and the a priori information of each fusion system in the Delegation play key roles in the laws which are generated by the Commissioner. These laws in turn, and in conjunction with the locally generated hypotheses, then define the types of information (neighborhood or nodal) which are communicated to peer fusion systems via the Delegate wrapper logic. The only requirement for a fusion process to be wrapped by the Delegate logic is that its inputs and outputs can be formally and structurally defined. For example, a rules-based, graph matching, or Bayesian network system could be wrapped while a Neural Network could not.

Each Delegate maintains a local multidimensional profile that is communicated to the Commissioner when changes occur or the fusion service comes online. The Commissioner maintains the *Instruction Control Matrix* as shown in Figure 18, which is the current state of every Delegate profile in the Delegation. The IntellEx server control logic's responsibility is to analyze this Instruction Control Matrix and determine, based on intersections in Delegate profile dimensions, the Collaboration Instructions to be shared with each Delegate. These instructions sets are specific to each Delegate and define instructions on which information to share, at what times, and with which Delegates. The Commissioner / Delegate divide in functionality is not a master / slave relationship. The Commissioner sees no runtime information from the Delegate wrapped fusion services and thus cannot generate very low level instructions. The instructions provided by the Commissioner are at a high level and dynamic communication instructions which are conditional on local fusion service estimate states are processed within the Delegate giving it more autonomy and requiring more intelligent client side logic.

A Delegate's role within IntellEx is much like an adapter within an enterprise service bus (ESB). ESB adapters can be said to provide syntactic abstraction of application logic above the underlying communication protocols being used. Delegates are not meant to replace this type of adapter, but rather compliment it. Delegate's can be said to provide **semantic** abstraction of application logic above the underlying information being transmitted across the ESB. The role of the Delegate is to adapt local information into global information for use within other peer fusion services. The job of the Delegate is to identify information which would be globally relevant regardless of whether it is locally significant or not. For SOA goals to be fully realized this type of semantic abstraction is necessary. There have been many publications in the literature involving the collaboration between agents [25][26]. Most of this work, however, has been on the standardization of communication syntax and not on how decisions on information sharing can be made. Our research can take advantage of these standards such as Knowledge Interchange Format (KIF) and the Knowledge Query and Manipulation Language (KQML) as syntactic formats for information payloads.



(a) Template graph matches in data graph

(b) Neighborhood of matches in data graph

Figure 19 - TruST Algorithm

"Neighborhood" is a word with many different levels of meaning in mathematics. One of the most general concepts of the neighborhood of a point *x* in (R^n) ϵ-neighborhood or infinitesimal open set, is the set of points inside an *n*-ball with center *x* and radius $\epsilon>0$. A set containing an open neighborhood is also called a neighborhood. The neighborhood of a vertex *v* in a graph is the set of all the vertices adjacent to *v*. More generally, the i^{th} neighborhood of *v* is the set of all vertices that lie at the distance *i* from *v*.

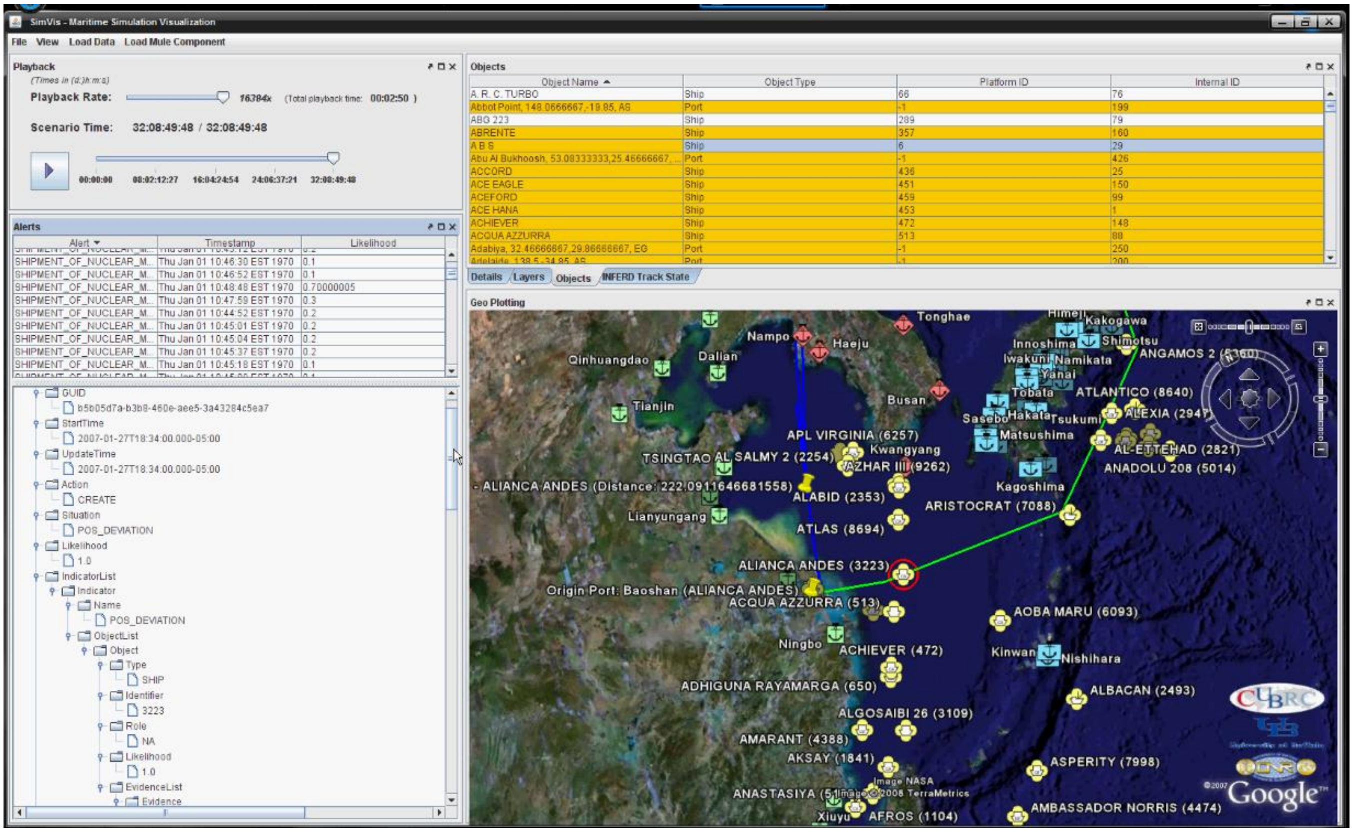
In the mathematical area of graph theory, the neighborhood of a vertex *v* in a graph *G* is the induced subgraph of *G* consisting of all vertices adjacent to *v* and all edges connecting two such vertices. Two vertices *u* and *v* are considered adjacent if an edge exists between them. This is denoted by $u{\downarrow}v$. The neighborhood [27] is often denoted $N_G(v)$ or (when the graph is unambiguous) $N(v)$. The same neighborhood notation may also be used to refer to sets of adjacent vertices rather than the corresponding induced subgraphs. The neighborhood described above does not include *v* itself, and is more specifically the open neighborhood of *v*; it is also possible to define a neighborhood in which *v* itself is included, called the closed neighborhood and denoted by $N_G[v]$. When stated without any qualification, a neighborhood is assumed to be open. In a topological space *S*, an open neighborhood of a point *x* is an open set containing *x*. A set containing an open neighborhood is simply called a neighborhood. The subscript *G* is usually dropped when there is no danger of confusion; the same neighborhood notation may also be used to refer to sets of adjacent vertices rather than the corresponding induced subgraphs. For a simple graph, the number of neighbors that a vertex has coincides with its degree.

If all vertices in *G* have neighborhoods that are isomorphic to the same graph *H*, *G* is said to be locally *H*, and if all vertices in *G* have neighborhoods that belong to some graph family *F*, *G* is said to be locally *F*. Neighborhoods are also used in the clustering coefficient of a graph, which is a measure of the average density of its neighborhoods. In [28], authors have presented a neighborhood broadcast and gossiping problem. For neighborhood broadcast the authors have considered nodes which are one edge away and for gossiping problem they have considered nodes, which are accessible from the main node. Schenker et al. [29] have compared a vector based graph representation, combined with a k-Nearest Neighbor (k-NN) algorithm to the graph matching approach, to represent web documents. There is a limited amount of literature available in the field of neighborhood structure, but we have found the concept of neighborhood of nodes used in various fields to calculate distances, get density of graphs, etc.

We are in the process of applying the TRUncated Search Tree algorithm (TRUST) [30], shown in Figure 6, to discover local neighborhood information which would be relevant to peer Delegates. Based on the settings for the algorithm parameters, we get various matches for the given template (local hypothesis) to peer Delegate profiles. The matches are clustered using K-means clustering, to find the most significant information among them. But this information is limited to the template we are trying to match. Consider a terrorist template which is matched against a given social network represented as data graph. The matches of terrorist template represent as most plausible terrorist networks in the given data graph. Having this information we have no idea as to what the terrorists are planning. The neighborhood of the matches can give information on the activities and targets the terrorist group perceives. So the neighborhood of a match is as important as the match. To get more information about the neighborhood of the matches an algorithm is developed to find the neighborhood score for each match in a data graph.

5.1.2 Develop Prototype Software that Implements H^{2}LIFT Architecture and Algorithms

Requirement Text: The Contractor shall develop a software prototype that implements the architecture and algorithms developed under 5.1.1. The software developed for this task will be written in Java to allow for interoperability with other systems, such as input managers and visualization tools. The software will also be adapted to sensitivity analysis, the output from which will be used during the performance evaluation of the algorithms.



Playback

(Times in (d:)h:m:s)

Playback Rate: 16384x (Total playback time: 00:02:50)

Scenario Time: 32:08:49:48 / 32:08:49:48

00:00:00 08:02:12:27 16:04:24:54 24:06:37:21 32:08:49:48

Objects

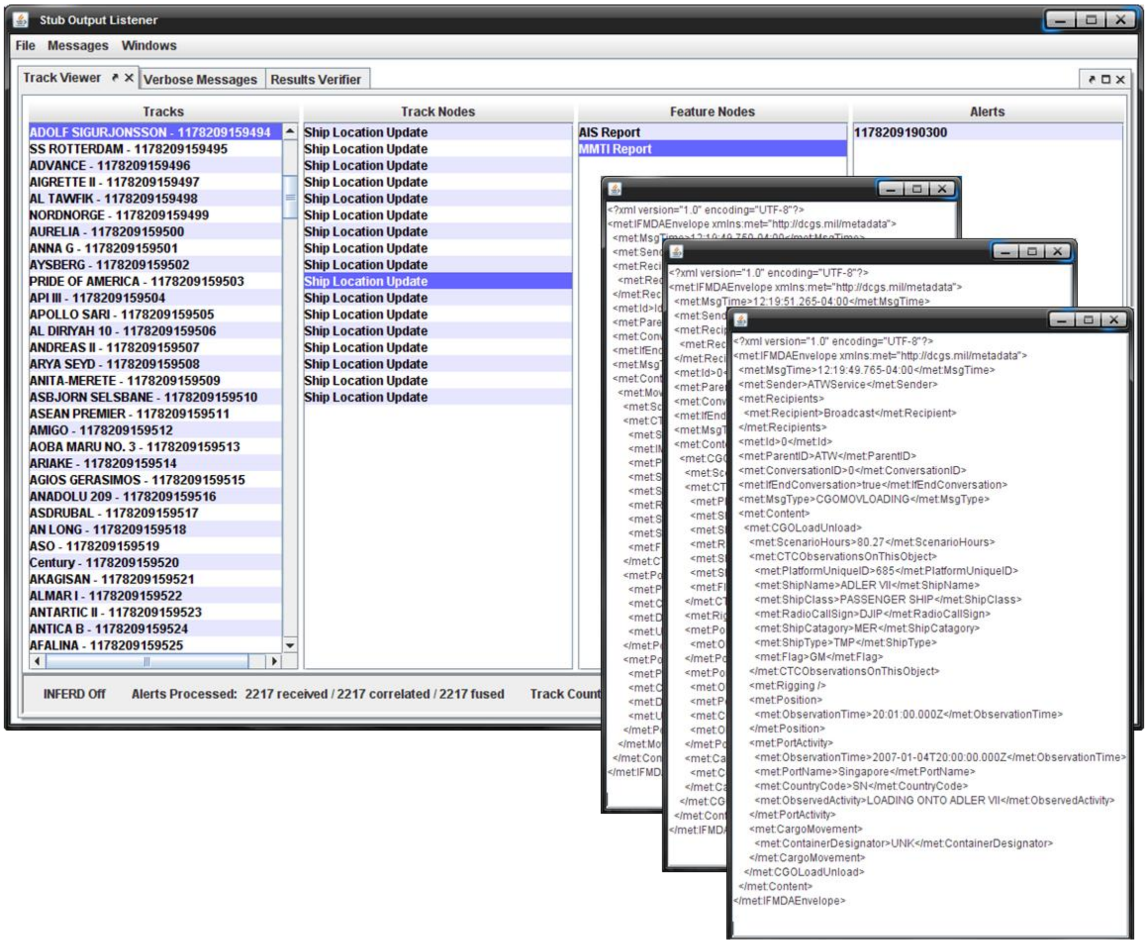
Object Name	Object Type	Platform ID	Internal ID
A. R. C. TURBO	Ship	66	76
Abbot Point, 148.0666667,-19.85, AS	Port	-1	199
ABG 223	Ship	289	79
ABRENTE	Ship	357	160
A B S	Ship	6	29
Abu Al Bukhoosh, 53.08333333,25.46666667, ...	Port	-1	426
ACCORD	Ship	436	25
ACE EAGLE	Ship	451	150
ACEFORD	Ship	459	99
ACE HANA	Ship	453	1
ACHIEVER	Ship	472	148
ACQUA AZZURRA	Ship	513	88
Adabiya, 32.46666667,29.86666667, EG	Port	-1	250
Adelaide, 138.5,-34.85, AS	Port	-1	200

Alerts

Alert	Timestamp	Likelihood
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:45:12 EST 1970	0.2
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:46:30 EST 1970	0.1
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:46:52 EST 1970	0.1
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:48:48 EST 1970	0.70000005
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:47:59 EST 1970	0.3
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:44:52 EST 1970	0.2
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:45:01 EST 1970	0.2
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:45:04 EST 1970	0.2
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:45:37 EST 1970	0.2
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:45:18 EST 1970	0.1
SHIPMENT_OF_NUCLEAR_M...	Thu Jan 01 10:45:00 EST 1970	0.1

```
GUID
  b5b05d7a-b3b8-460e-aee5-3a43284c5ea7
StartTime
  2007-01-27T18:34:00.000-05:00
UpdateTime
  2007-01-27T18:34:00.000-05:00
Action
  CREATE
Situation
  POS_DEVIATION
Likelihood
  1.0
IndicatorList
  Indicator
    Name
      POS_DEVIATION
    ObjectList
      Object
        Type
          SHIP
        Identifier
          3223
        Role
          NA
        Likelihood
          1.0
        EvidenceList
          Evidence
```

Figure 20 - Screen Shot of MDA Visualization



Tracks	Track Nodes	Feature Nodes	Alerts
ADOLF SIGURJONSSON - 1178209159494	Ship Location Update	AIS Report	1178209190300
SS ROTTERDAM - 1178209159495	Ship Location Update	MMTI Report	
ADVANCE - 1178209159496	Ship Location Update		
AIGRETTE II - 1178209159497	Ship Location Update		
AL TAWFIK - 1178209159498	Ship Location Update		
NORDNORGE - 1178209159499	Ship Location Update		
AURELIA - 1178209159500	Ship Location Update		
ANNA G - 1178209159501	Ship Location Update		
AYSBERG - 1178209159502	Ship Location Update		
PRIDE OF AMERICA - 1178209159503	Ship Location Update		
API III - 1178209159504	Ship Location Update		
APOLLO SARI - 1178209159505	Ship Location Update		
AL DIRIYAH 10 - 1178209159506	Ship Location Update		
ANDREAS II - 1178209159507	Ship Location Update		
ARYA SEYD - 1178209159508	Ship Location Update		
ANITA-MERETE - 1178209159509	Ship Location Update		
ASBJORN SELSBANE - 1178209159510	Ship Location Update		
ASEAN PREMIER - 1178209159511			
AMIGO - 1178209159512			
AOBA MARU NO. 3 - 1178209159513			
ARIAKE - 1178209159514			
AGIOS GERASIMOS - 1178209159515			
ANADOLU 209 - 1178209159516			
ASDRUBAL - 1178209159517			
AN LONG - 1178209159518			
ASO - 1178209159519			
Century - 1178209159520			
AKAGISAN - 1178209159521			
ALMAR I - 1178209159522			
ANTARTIC II - 1178209159523			
ANTICA B - 1178209159524			
AFALINA - 1178209159525			

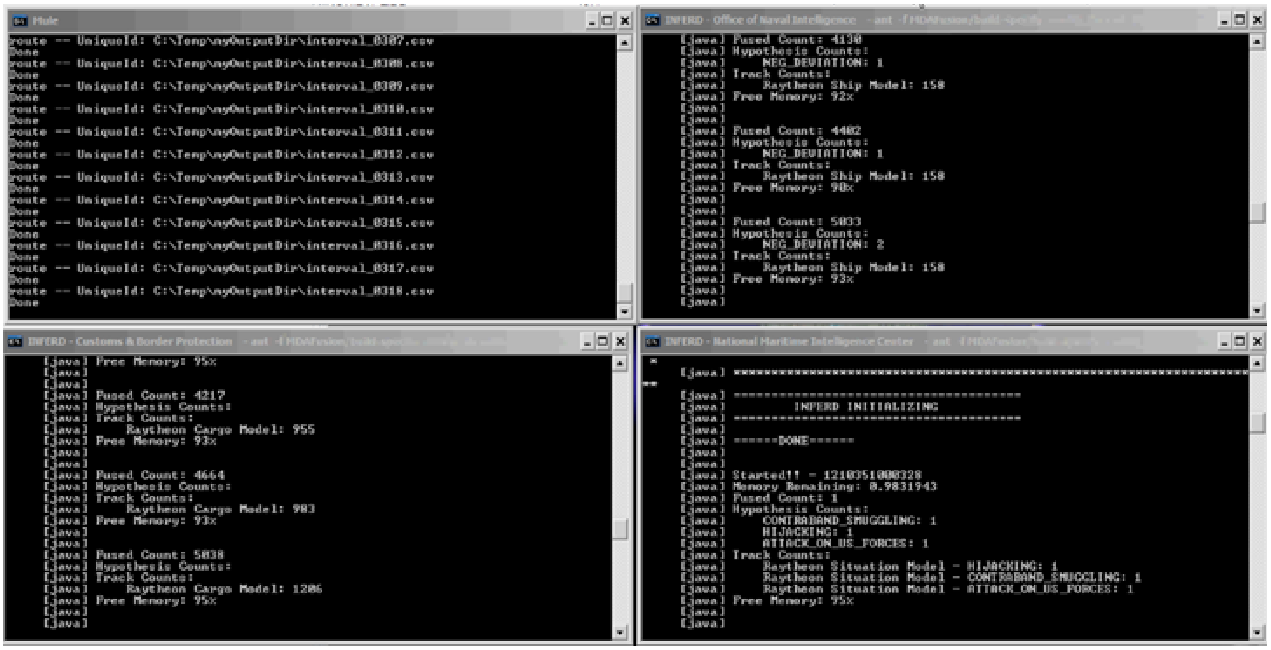
INFERD Off   Alerts Processed: 2217 received / 2217 correlated / 2217 fused   Track Count

```
<?xml version="1.0" encoding="UTF-8"?>
<met:IFMDAEnvelope xmlns:met="http://dcgs.mil/metadata">
```

```
<?xml version="1.0" encoding="UTF-8"?>
<met:IFMDAEnvelope xmlns:met="http://dcgs.mil/metadata">
  <met:MsgTime>12:19:51.265-04:00</met:MsgTime>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<met:IFMDAEnvelope xmlns:met="http://dcgs.mil/metadata">
  <met:MsgTime>12:19:49.765-04:00</met:MsgTime>
  <met:Sender>ATWService</met:Sender>
  <met:Recipients>
    <met:Recipient>Broadcast</met:Recipient>
  </met:Recipients>
  <met:Id>0</met:Id>
  <met:ParentID>ATW</met:ParentID>
  <met:ConversationID>0</met:ConversationID>
  <met:IfEndConversation>true</met:IfEndConversation>
  <met:MsgType>CGOMOVLOADING</met:MsgType>
  <met:Content>
    <met:CGOLoadUnload>
      <met:ScenarioHours>80.27</met:ScenarioHours>
      <met:CTCObservationsOnThisObject>
        <met:PlatformUniqueID>685</met:PlatformUniqueID>
        <met:ShipName>ADLER VII</met:ShipName>
        <met:ShipClass>PASSENGER SHIP</met:ShipClass>
        <met:RadioCallSign>DJIP</met:RadioCallSign>
        <met:ShipCatagory>MER</met:ShipCatagory>
        <met:ShipType>TMP</met:ShipType>
        <met:Flag>GM</met:Flag>
      </met:CTCObservationsOnThisObject>
      <met:Rigging />
      <met:Position>
        <met:ObservationTime>20:01:00.000Z</met:ObservationTime>
      </met:Position>
      <met:PortActivity>
        <met:ObservationTime>2007-01-04T20:00:00.000Z</met:ObservationTime>
        <met:PortName>Singapore</met:PortName>
        <met:CountryCode>SN</met:CountryCode>
        <met:ObservedActivity>LOADING ONTO ADLER VII</met:ObservedActivity>
      </met:PortActivity>
      <met:CargoMovement>
        <met:ContainerDesignator>UNK</met:ContainerDesignator>
      </met:CargoMovement>
    </met:CGOLoadUnload>
  </met:Content>
</met:IFMDAEnvelope>
```

Figure 21 - Screen Shot of INFERD Tracks


Mule
INFERD - Office of Naval Intelligence - ant -f MDAFusion/build
INFERD - Customs & Border Protection - ant -f MDAFusion/build
INFERD - National Maritime Intelligence Center - ant -f MDAFusion/build

```
route -- UniqueId: C:\Temp\myOutputDir\interval_0307.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0308.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0309.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0310.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0311.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0312.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0313.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0314.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0315.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0316.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0317.csv
Done
route -- UniqueId: C:\Temp\myOutputDir\interval_0318.csv
Done
```

```
     [java] Fused Count: 4130
     [java] Hypothesis Counts:
     [java]      NEG_DEVIATION: 1
     [java] Track Counts:
     [java]      Raytheon Ship Model: 158
     [java] Free Memory: 92%
     [java]
     [java]
     [java] Fused Count: 4402
     [java] Hypothesis Counts:
     [java]      NEG_DEVIATION: 1
     [java] Track Counts:
     [java]      Raytheon Ship Model: 158
     [java] Free Memory: 90%
     [java]
     [java]
     [java] Fused Count: 5033
     [java] Hypothesis Counts:
     [java]      NEG_DEVIATION: 2
     [java] Track Counts:
     [java]      Raytheon Ship Model: 158
     [java] Free Memory: 93%
     [java]
     [java]
```

```
     [java] Free Memory: 95%
     [java]
     [java]
     [java] Fused Count: 4217
     [java] Hypothesis Counts:
     [java] Track Counts:
     [java]      Raytheon Cargo Model: 955
     [java] Free Memory: 93%
     [java]
     [java]
     [java] Fused Count: 4664
     [java] Hypothesis Counts:
     [java] Track Counts:
     [java]      Raytheon Cargo Model: 983
     [java] Free Memory: 93%
     [java]
     [java]
     [java] Fused Count: 5038
     [java] Hypothesis Counts:
     [java] Track Counts:
     [java]      Raytheon Cargo Model: 1206
     [java] Free Memory: 95%
     [java]
     [java]
```

```
 *
     [java] ********************************************************************
**
     [java] =======================================
     [java]          INFERD INITIALIZING
     [java] ---------------------------------------
     [java]
     [java] ======DONE======
     [java]
     [java]
     [java] Started!! - 1210351000328
     [java] Memory Remaining: 0.9831943
     [java] Fused Count: 1
     [java] Hypothesis Counts:
     [java]      CONTRABAND_SMUGGLING: 1
     [java]      HIJACKING: 1
     [java]      ATTACK_ON_US_FORCES: 1
     [java] Track Counts:
     [java]      Raytheon Situation Model - HIJACKING: 1
     [java]      Raytheon Situation Model - CONTRABAND_SMUGGLING: 1
     [java]      Raytheon Situation Model - ATTACK_ON_US_FORCES: 1
     [java] Free Memory: 95%
     [java]
     [java]
```

Figure 22 - Screen Shot of 4 Running Fusion Node Consoles

5.1.3 Develop an Integrated Simulation Tool for System Performance Evaluation and Analysis

Requirement Text: The Contractor shall develop a dynamic, scenario-based tool that will be used to generate data that will be used to support the development of the IF algorithms and to provide data for performance evaluation and analysis.

5.1.3.1 Brief Overview

The CUBRC Contextual Simulator was created based on the need for the ability to track and monitor events occurring in the Maritime Domain. The simulator takes in a set a pre-determined knowledge files, and logically generates a scenario, or a set of comma separated data files. These files contain messages that describe observed activities in or about the Maritime Domain. The output files will then serve as input to a fusion engine.

The user can enter custom messages that pertain to the specific scenario being generated. This gives the user the ability to control certain events happening in the output data so they can create their own scenario or replicate a white paper scenario.

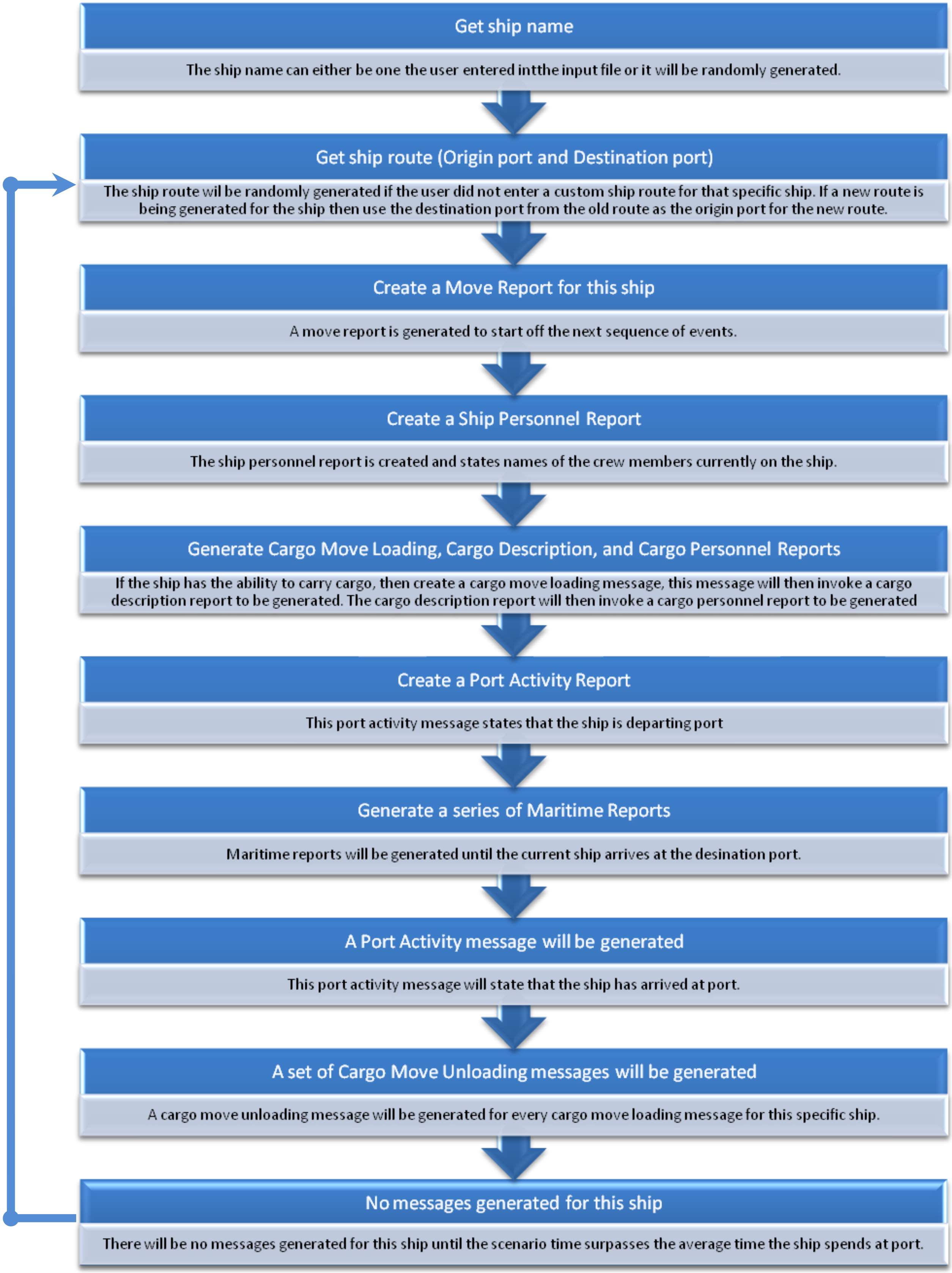
5.1.3.2 General Logic & Sequence of Execution

The simulator reads in a configuration file in order to be able to read in the set of knowledge files. The simulator will then build corresponding data structures to make the search for information more efficient.

The message generation simulates the idea of having a satellite be able to take a picture of the earth at a user defined interval. In the default configuration, this interval is set to 5 hours. When an interval is reached, a series of messages will be generated for each ship the scenario includes. When a message is being generated, the simulator will first check the user input file to see if the user has defined a custom message. If there is a user defined message, then that message will be written to the output file. If not then the simulation will randomly generate each field for the specific type of message that is being created.

When all of the messages are created for that specific ship during that time interval, the simulator will repeat those steps for the next ship. Once the messages for all of the ships have been generated, the scenario hours are increased by the defined interval and the simulation process starts over.

An example sequence of events for one ship is presented in Figure 23.



Get ship name

The ship name can either be one the user entered intthe input file or it will be randomly generated.

Get ship route (Origin port and Destination port)

The ship route wil be randomly generated if the user did not enter a custom ship route for that specific ship. If a new route is being generated for the ship then use the destination port from the old route as the origin port for the new route.

Create a Move Report for this ship

A move report is generated to start off the next sequence of events.

Create a Ship Personnel Report

The ship personnel report is created and states names of the crew members currently on the ship.

Generate Cargo Move Loading, Cargo Description, and Cargo Personnel Reports

If the ship has the ability to carry cargo, then create a cargo move loading message, this message will then invoke a cargo description report to be generated. The cargo description report will then invoke a cargo personnel report to be generated

Create a Port Activity Report

This port activity message states that the ship is departing port

Generate a series of Maritime Reports

Maritime reports will be generated until the current ship arrives at the desination port.

A Port Activity message will be generated

This port activity message will state that the ship has arrived at port.

A set of Cargo Move Unloading messages will be generated

A cargo move unloading message will be generated for every cargo move loading message for this specific ship.

No messages generated for this ship

There will be no messages generated for this ship until the scenario time surpasses the average time the ship spends at port.

Figure 23 - MDA Simulator Information Flow

5.1.3.3 Contextual Input Data Sources

The following set of files includes all of the necessary contextual information needed to generate the contextual-dependant message types. These files were designed to contain real world data to make the simulator produce a more authentic output. The knowledge files can be easily altered to contain whatever information is needed by simply adding or deleting content.

Any data that was unavailable for use or did not openly exist was simulated through the extrapolation of existing data. The abundant amount of data is a cohesive collection of information concerning Maritime Domain Awareness (MDA.) These files will be referred to as the simulator's "knowledge files" as they are the source at which contextual information is drawn. They serve as constraints to the simulator as it generates its outputs based on the inputs it receives through these knowledge files.

UserInputFile.xls and *MDAContextualConfiguration.xml* are two user-configurable files which defines custom scenarios and outputs.

Bank List.csv

- Provides 30,000+ banks that are associated with a specific country/territory.
- Source:
    - http://en.wikipedia.org/wiki/List_of_banks
        - Banks were manually handpicked and assigned an OTH-Gold country code.
        - 98.46% of the data is actual real world data.
        - The additional data is simulated to provide a complete data set. For countries and regions without a central bank, the simulated bank names represent foreign bank influences in that region.

PersonListFirstLast.csv

- Provides first/last names for American, Japanese, Chinese, Italian, French, Spanish, English, Arabic, and Russian nationalities.
- Each nationality has an associated column with first names and another for last names.
- The simulator picks a first name and randomly matches it to a respective last name. This allows for a more extensive set of unique names.
- Sources:
    - http://nine.frenchboys.net/
    - http://en.wikipedia.org/wiki/Most_popular_given_names

PortListNewDecimal.csv

- Collection of nearly 5,000 ports throughout the world with its assigned country and respective country code.
- A lat/long is provided for its location, accurate to the nearest minute.
- United Nations Conference on Trade and Development (UNCTAD) codes are provided for each port. This code is a unique identifier assigned to a port by a special organization of the United Nations.
- Additional data provided includes the status of the port, its max draft, and any other ports that it may be associated with.
- Source:
    - http://www.e-ships.net/ports.php

CompanyListBySITC.csv

- Provides over 78,000 company names categorized by Standard International Trade Classification (SITC) code. Each SITC code is used to classify goods internationally imported and exported.
- The company names are associated with a SITC code so that the simulator can correlate companies based on the type of goods being shipped and vice versa.
- Source:
  - http://www.alibaba.com/archives/company/0/companies.html
    - Alibaba.com® provides actual company names sorted by the type of goods they provide.
    - All 78,000+ companies were manually acquired.

TransportCompanyNames.csv

- List of both shipping and ground transport companies.
- Source:
  - http://www.alibaba.com/archives/company/0/companies.html

PlatformCharacteristics.csv

- List of each ship class involved within the domain of the simulator.
- Each ship class has associated characteristics including average speed, average hours in port, average crew sizes, and relevant SITC codes for that ship along with additional data.

ShipList.csv

- Provides a list of nearly 10,000 ships used in the contextual generator.
- Each ship has an International Maritime Organization (IMO) number for unique identification purposes. Each IMO number references a ship name, its call sign, the ships type, its gross weight tonnage, the date in which it began operating, and its flag along with its associated country code.
- By using both the *ShipList.csv* and the *PlatformCharacteristics.csv* file, a more complex ship profile will be built.

UserInputFile.xls

- Provides the user the ability to enter in custom messages that will be included in the output files.
- The custom fields the user can enter are:
  - Ship Names
  - Ship Routes
  - Cargo Move Loading Messages
  - Cargo Description Messages
  - Cargo Personnel Reports
  - Ship Personnel Reports
  - Financial Transaction Reports

MDAContextualSimConfiguration.xml

- Contains the file names and locations of all the input files.
- Allows the user to enter Global Parameters for the simulation
  - Scenario Length
  - Amount of ships
  - Scenario time interval
  - Number of cargo messages per ship

5.1.3.4 Message Types

Each message type provides a scenario with an array of data, each containing a unique subset of relevant information. The first three messages (MarRep, PortAct, MOVREP) contain information about the current and planned location of individual ships. The CGOMOVLoading and CGOMOVUnloading messages are used to uniquely identify a piece of cargo.

The remaining message types are all contextually based (CGODescrip, CGOPERSONRep, SHIPPERSONRep, TRANSRep) and provide for a full-bodied scenario. Information is selectively drawn from the knowledge files and is used to create individual messages. These messages are exclusively constrained to those files which already have a substantial amount of data to draw from.

Maritime Report (MarRep)

- A Maritime report specifies the position and observations on a vessel
  - Current ship Latitude
  - Current ship Longitude
  - Destination
- A MarRep should be generated anytime a ship is sensed when it is in open water

Port Activity (PortAct)

- A Port Activity is an observed activity at a port
  - States whether a ship is currently departing the port or arriving at the port
- This message is generated whenever a ship notifies the port they are departing or arriving.

Movement Report (MOVREP)

- A Movement report is planned movement for a vessel per LLOYDS
  - States the ships Origin port and Destination port
  - Provides an estimate time of when the ship should arrive at its destination
- This message is usually the first message in a sequence of messages generated for a ship. States that the ship preparing to depart.

Cargo Move Loading (CGOMOVLoading)

- A Cargo Move Loading message is an observed movement of cargo from a port to a vessel.
  - Contains the cargo destination
  - SITC Code for the contents of the container
  - Container Identifier number
  - Port where the observation occurred
- This message usually occurs after a Movement Report, but before a Port Activity.

Cargo Move Unloading (CGOMOVUnloading)

- A Cargo Move Unloading message is an observed movement of cargo from a vessel to a port.
  - SITC Code for the contents of the container
  - Container Identifier number
  - Port where the observation occurred
- This message is generated once there has been a Port Activity message stating that the ship has arrived at port.

Cargo Description (CGODescrip)

- A Cargo Description message is information about the contents of the container.
  - SITC Code for the contents of the container
  - Contents specified in the container
  - Cargo Destination
  - Origin of the contents (Company Factory)
  - Containers tamper proof information (Bolt Seal Serial Number, eSeal Code etc.)
  - Ship that cargo is on
  - Cargo Inspection Information
- One of these messages are generated every time a Cargo Move Loading message is generated

Cargo Personnel Report (CGOPERSONRep)

- A Cargo Personnel Report is information about who came in contact with the cargo.
  - Container Identifier Number
  - Place the container was stuffed
  - Where the container was inspected
  - Names of Stuffing Crew
  - Names of Loading Crew
  - Name of container sealer
- This message is generated for every Cargo Description message.

Ship Personnel Report (SHIPPERSONRep)

- A Ship Personnel Report is information about the people on or involved with a certain ship.
  - Ship name and destination
  - Ship crew members names
  - Ship Owner name
- This message is generated after a Move Report. Once the ship states that it is going to leave port.

Financial Transaction Report (TRANSRep)

- A Financial Transaction Report is information about a financial transaction that has been flagged.
  - Type of Transaction (Deposit or Withdrawal)
  - Person's Name
  - Place of Transaction
  - Amount of the transaction
- This message can occur at anytime throughout the scenario.

5.1.3.5 V. Output Files

The generated messages are sequentially written to a Comma Separated File. Each of these files contains 1000 messages, or 1000 lines of data. Once the current file contains 1000 lines of data, it is closed out and a new file is made. The output files are named in order of creation. For example, the first file will be called interval_0.csv and the second file will be called interval_1.csv.

These output files will in turn be used as the input for the fusion engine. Because the user has the ability to specify certain things to happen at certain times, they will be able to create anomalies in the output data. Therefore this data will be useful when testing the fusion engine.

5.2 Deliverables

5.2.1 Architecture and Algorithms for GWOT/MDA via Interim Technical Reports

Completed. Reference Section 5.1 for a discussion of outcomes.

5.2.2 Prototype Software for the implementation of H^{2}LIFT architecture and algorithms (includes source and executable codes)

Completed. The architecture and algorithms referenced in Section 5.2.1 have been coded and tested using data generated from task 5.2.3. All source code has been packaged on a CD accompanying this final report.

5.2.3 Simulation Tool: MATLAB/SIMULINK (includes source and executable codes delivered on CD)

Completed. The Simulator has been prototyped in MATLAB and later rewritten in the Java programming language for facilitated distribution and ease of 3rd party integration. All documentation and code has been packaged on a CD accompanying this final report.

5.2.4 User's Manual (Delivered on CD)

Completed. The user manual describing simulated items and required input has been included both on CD with the simulator source code and executables and within this report in Section 5.1.3.

5.2.5 Kickoff Meeting and Program Reviews

All program relevant reviews and meetings have been attended. Accompanying this report on CD are a copy of briefings given at each of these reviews.

5.2.6 Quarterly Progress Reports

All program relevant quarterly progress reports have been submitted in compliance with this contract deliverable. Accompanying this report on CD are a copy of the reports which have been submitted.

5.2.7 Final Technical Report

This document is submitted in fulfillment of this contract deliverable.

6 CDRL STATUS

The following table describes the status of the Contract Data Requirements List (CDRL) identified in the referenced contract.

Data Item No.	**Data Item Title**	**Status**
A001	Interim Technical Reports	Completed.
A002	Progress Reports	Completed.
A003	Kickoff Meeting	Completed.
A004	Program Reviews	Completed.
A005	Final Report	Completed.